**Trabajo Final Master**

Autor: Martín López Scala

Tutor: José Miguel Hernández

Set de Datos elegido: https://www.kaggle.com/c/home-credit-default-risk/

**1. Introducción con el objetivo del análisis**

La idea para el TFM es la de generar modelos de análisis de riesgo para una cartera de préstamos del set de datos de Home Credit.

Para el trabajo se cuenta con 8 ficheros.

**application\_train / application\_test:** son los datos principales de capacitación y pruebas contienen información sobre cada solicitud de préstamo en Home Credit. Cada préstamo tiene su propia fila y se identifica mediante el campo SK\_ID\_CURR. Los datos de la solicitud del fichero application\_train vienen con el campo TARGET (OBJETIVO). Indican 0 (cero), para préstamos pagados y 1 (uno) para el que no se pagó.

**Oficina (bureau):** contiene datos sobre créditos anteriores del cliente en otras instituciones financieras. Cada crédito anterior tiene su propia fila en el fichero bureau, pero un préstamo en los datos de la aplicación (train) puede tener múltiples créditos anteriores. Este fichero incluye el campo SK\_ID\_CURR que lo relaciona con el fichero de entrenamiento y prueba (train/test)

**bureau\_balance:** contiene los datos mensuales sobre los créditos anteriores en bureau. Cada fila es un mes de un crédito anterior, y un solo crédito anterior puede tener varias filas, una para cada mes de la duración del crédito. Este fichero contiene el campo SK\_ID\_BUREAU que lo relaciona con el fichero bureau.

**anterior\_aplicación (previous\_application):** solicitudes anteriores para préstamos en Home Credit de clientes que tienen préstamos en los datos de la aplicación (train). Cada préstamo actual en los datos de la aplicación train puede tener múltiples préstamos anteriores. Cada aplicación anterior tiene una fila y se identifica mediante la función SK\_ID\_PREV. Este fichero incluye el campo SK\_ID\_CURR que lo relaciona con el fichero de entrenamiento y prueba (train/test)

**POS\_CASH\_BALANCE:** contiene datos mensuales sobre puntos de venta anteriores o préstamos en efectivo que los clientes han tenido con Home Credit. Cada fila es un mes de un punto de venta anterior o préstamo en efectivo y se identifica mediante la función SK\_ID\_PREV. Un solo préstamo anterior puede tener muchas filas. Este fichero incluye el campo SK\_ID\_CURR que lo relaciona con el fichero de entrenamiento y prueba (train/test)

**credit\_card\_balance:** contiene datos mensuales sobre tarjetas de crédito anteriores que los clientes han tenido con Home Credit. Cada fila es un mes de saldo de tarjeta de crédito y se identifica mediante la función SK\_ID\_PREV. Una sola tarjeta de crédito puede tener muchas filas. Este fichero incluye el campo SK\_ID\_CURR que lo relaciona con el fichero de entrenamiento y prueba (train/test)

**cuotas\_pagos:** historial de pagos de préstamos anteriores en Home Credit. Hay una fila por cada pago realizado y una fila por cada pago perdido y se identifica mediante la función SK\_ID\_PREV. Este fichero incluye el campo SK\_ID\_CURR que lo relaciona con el fichero de entrenamiento y prueba (train/test)

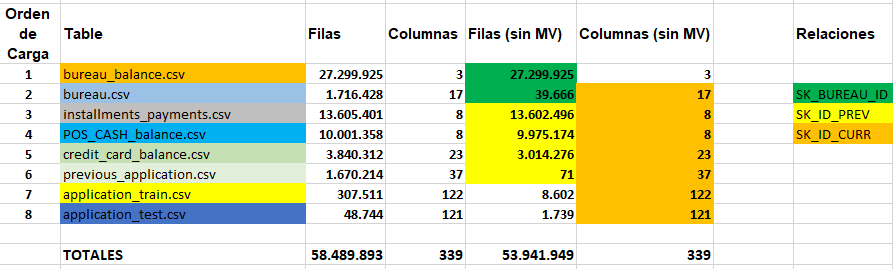
El objetivo sería determinar y agregar las características más relevantes de la información histórica de las solicitudes de préstamos contenida en estos ficheros al set de datos de entrenamiento (train) que contiene el campo objetivo para aplicar los modelos para predecir si un solicitante podrá pagar un préstamo o no.

Para la selección de estos campos en una primera instancia se tomara su correlación con respecto al TARGET a medida que se vayan uniendo al set de datos de entrenamiento (train)

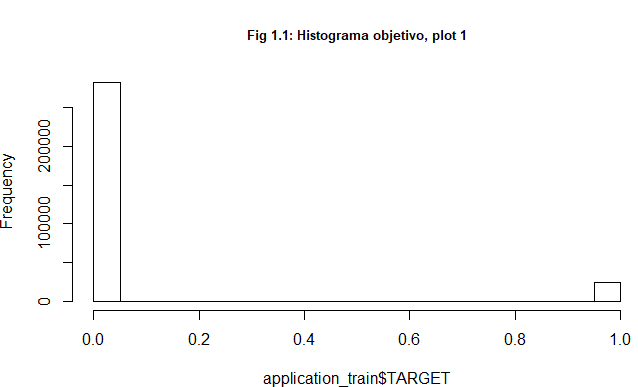
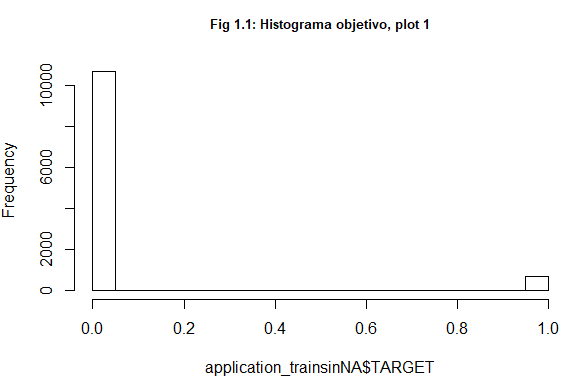
**2. Carga de los datos y análisis descriptivo**

Realizando la carga de datos en RStudio (va código adjunto) iniciamos un análisis exploratorio de los datos donde principalmente se vieron estadísticas y gráficos preliminares de las variables de cada fichero y se buscaron patrones o relaciones dentro de los datos.

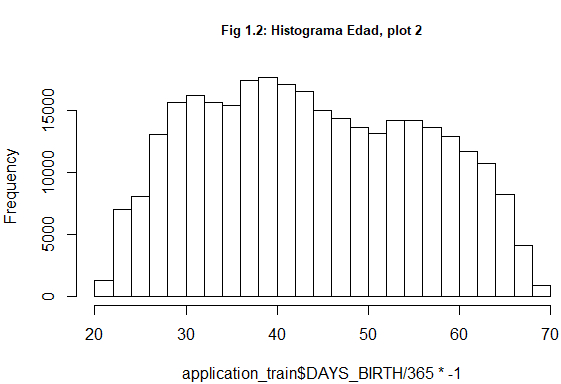
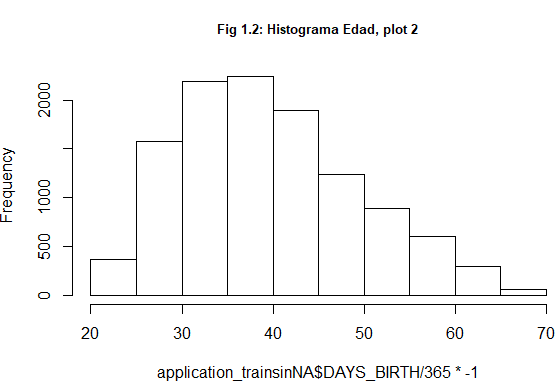
De aquí también armamos una planilla Excel (enviada en adjunto) con la identificación de los campos que componen cada fichero con una descripción del campo, tipo de dato y la distribución de datos en las variables categóricas. También agregamos un cuadro con descripción de filas y columnas por ficheros e impacto de los Missing Values (MV).



En cuanto a las variables se realizaron gráficos de distribución en el fichero train respecto al TARGET en el total de datos (application\_train$TARGET) y eliminando NA (application\_trainsinNA$TARGET). En principio no veríamos impacto de los NA pero si vemos que es una distribución desbalanceada. Los casos impagos (1) son bastante menores a los pagos (0).

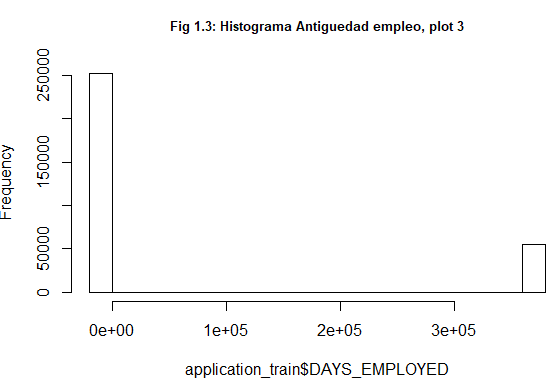
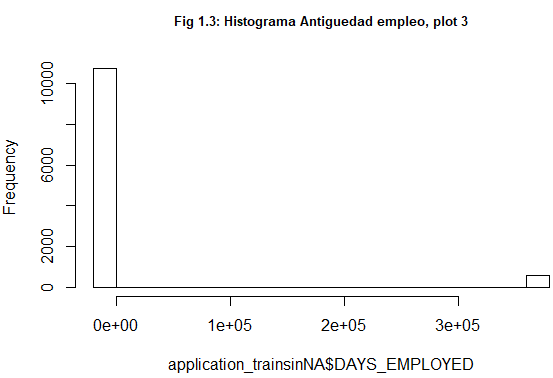
 

Otra de las variables vistas fue application\_train$DAYS\_BIRTH. Esta indica la edad del cliente en días al momento de la solicitud. Por ello se ajusto para llevar a años (se dividió en 365 y multiplicó por -1)

De aquí podemos decir que la edad de los titulares de los préstamos se concentra entre los 30 y los 50 años.

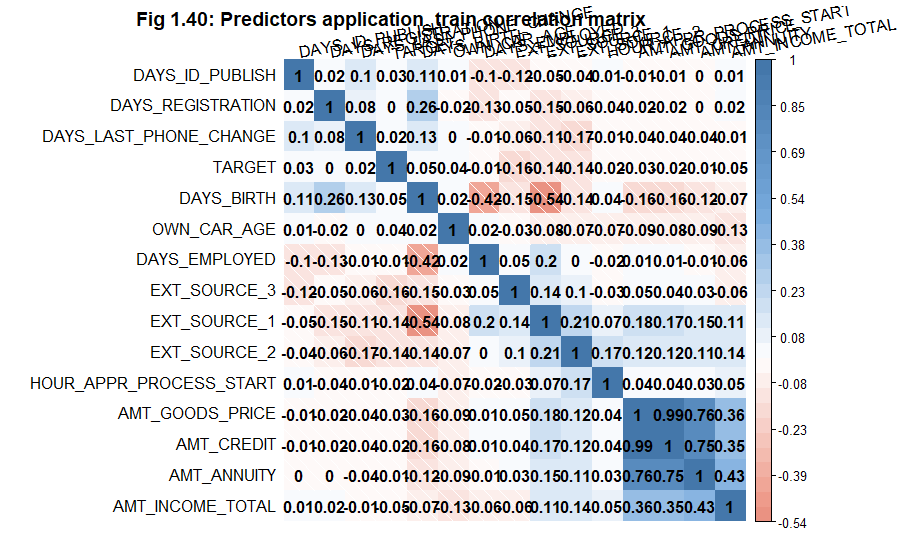
Otra de las variables vistas fue application\_train$DAYS\_EMPLOYED que indica ¿Cuántos días antes de la solicitud, la persona comenzó su empleo actual?

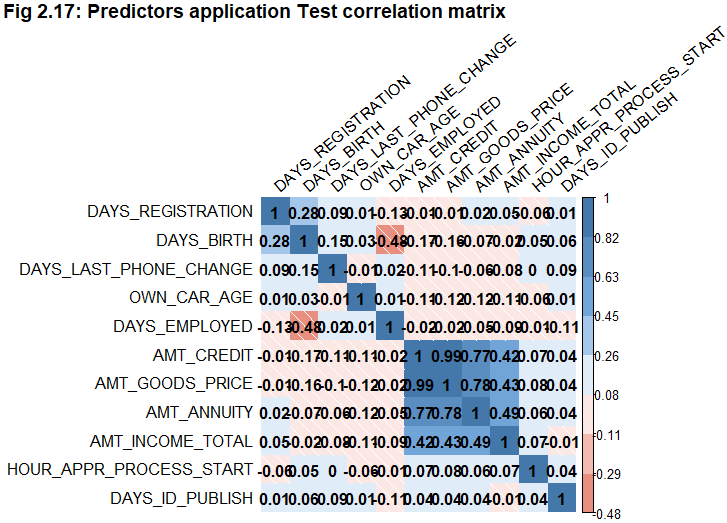
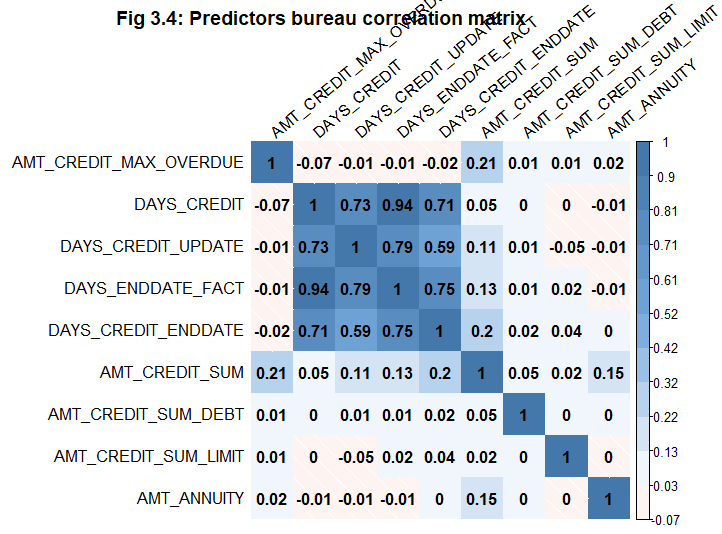
De estos gráficos se puede ver la existencia de valores extremos que están afectando la distribución.

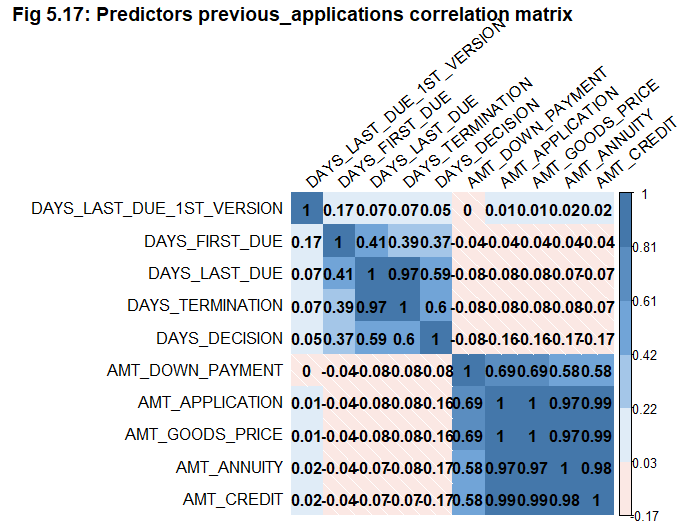
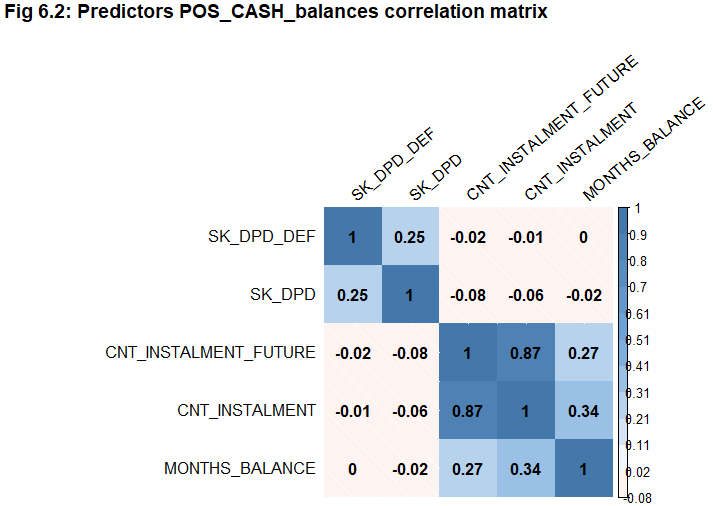
En cuanto a la correlación de las variables del fichero de entrenamiento (train), no se observan relaciones fuertes respecto al TARGET.

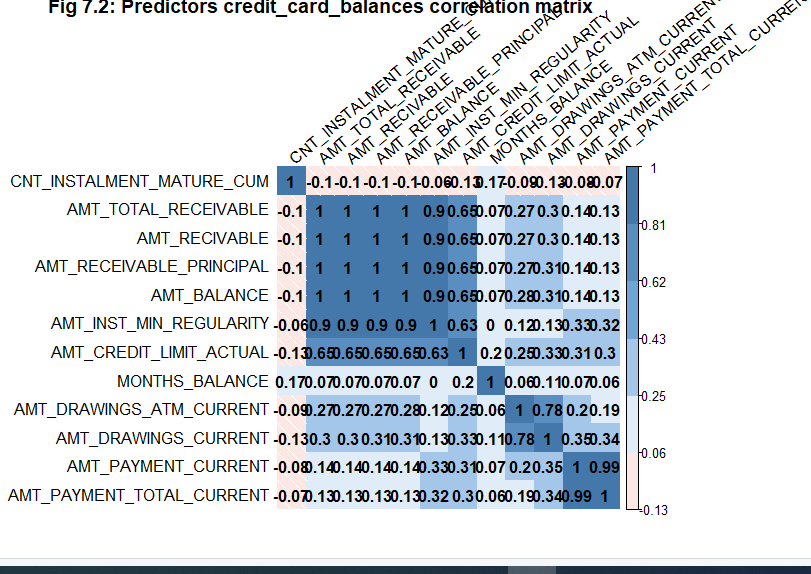
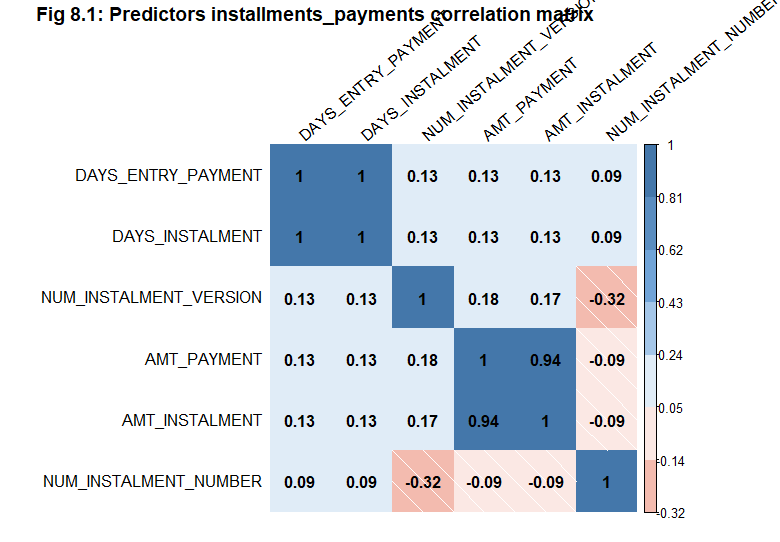
Si se observan relaciones fuertes entre las variables como AMT\_GOODS\_PRICE y AMT\_CREDIT por ejemplo pero todavía no se han analizado los valores extremos.



Del resto de los ficheros principalmente se observó, (y agrega al informe), las correlaciones entre variables para tener una referencia. Todas se consideraron sin valores NA y sobre variables numéricas.

**3. Selección y elección de variables**

Para la selección y elección de variables se comenzó a trabajar con un código en Python (notebook de Anaconda que se adjunta), donde fichero por fichero se fue realizando modificaciones hasta llegar al dataframe con las características que vamos a agregar al set de datos de entrenamiento (train).

Para la tarea se cargó la totalidad de los ficheros sin considerar los valores perdidos y las variables colineales. (correlación entre las variables).

Luego se aplicaron distintas funciones que definieron el trabajo y paso a describir.

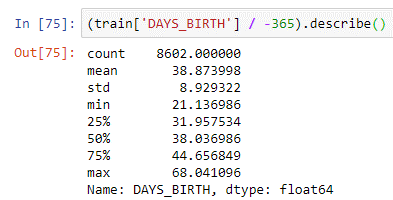
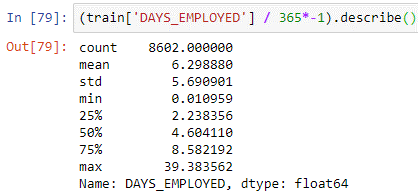
Comenzamos cargando el fichero bureau\_balance.csv y se generaron salidas para describir los datos contenidos

Para el manejo de las variables categóricas se aplicó una función que por cada característica del fichero agregaba su conteo y la media.

Para el manejo de las variables numéricas se aplicó una función que por cada característica del fichero agregaba su conteo, la media, el máximo, el mínimo y la suma.

Con esto atenuamos el impacto de los valores extremos.

Volvimos a verificar y acomodar los valores de los campos DAYS\_BIRTH y DAYS\_EMPLOYED del fichero train/test por las anomalías vistas en el análisis exploratorio en R

Recordemos que el fichero bureau\_balance.csv contiene los datos mensuales sobre los créditos anteriores del fichero bureau.

Primero se hizo el agregado de las variables categóricas por el campo SK\_ID\_BUREAU (en DF BureauBalance\_counts)

Segundo se hizo el agregado de las variables numéricas por el campo SK\_ID\_BUREAU (en DF BureauBalance\_agg)

Tercero se hizo el merge o fusion de los DF BureauBalance\_agg y BureauBalance\_counts por el campo SK\_ID\_BUREAU (en DF BureauBalance\_by\_loan)

Cuarto se hizo el merge (o fusión) de los DF BureauBalance\_agg y Bureau para agregar el campo SK\_ID\_CURR (DF BureauBalance\_by\_loanMerge) y en otro paso se eliminó el campo SK\_ID\_BUREAU quedando el DF **BureauBalance\_by\_client** que agregaremos al fichero train.

Para el fichero bureau primero se contaron los prestamos previos por campo SK\_ID\_BUREAU (DF Previous\_loan\_counts, se agregaron conteos y datos estadísticos a las variables numéricas (DF Bureau\_agg), se agregaron conteos y media para las variables categóricas (DF categorical\_grouped) y se realizó el merge (o fusión) de estos en el DF **BureauAgregado\_by\_loan** que se unirá a nuestro fichero train.

Para los ficheros installments\_payments.csv, POS\_CASH\_balance.csv, credit\_card\_balance.csv, previous\_application.csv primero se agrupo por campo SK\_ID\_PREV y luego se asignó al campo SK\_ID\_CURR. Luego se les hizo el agregado para las variables numéricas y categóricas. Los DF finales que se agregaron a train son **installments\_payments\_by\_client,** **POS\_CASH\_balance\_by\_client,** **credit\_card\_balance\_by\_client,** **previous\_application\_ag**

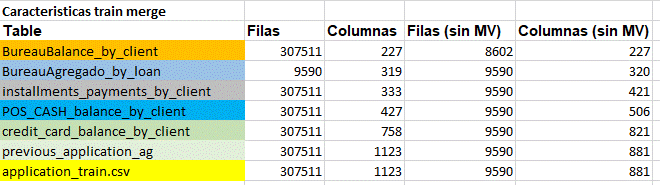
Vale mencionar que para el procesado de la información de los ficheros también se consideró el análisis de la correlación entre variables, la estadística de los Missing values y la posibilidad de generar la carga DF sin los Missing values.

Respecto a ello se decidió no eliminar ni ajustar el set hasta que se realice la fusión con el fichero train y evaluar su impacto. De esta forma también podemos observar analizar las distintas variables y su correlación respecto al TARGET (Objetivo).

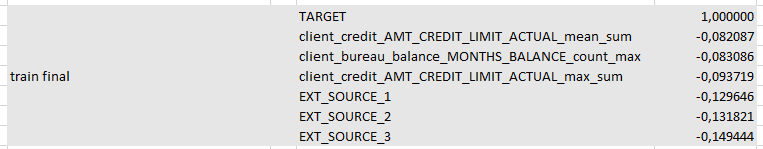
Se carga el fichero de entrenamiento (train), se calcula la correlación de las características que contiene respecto al TARGET y se genera una salida ordenando las mismas en función de su fuerza (positiva o negativa).

A partir de aquí entonces lo mencionado. Se realiza un merge de cada uno de los ficheros con el fichero train y se observa la variación sufrida en la correlación de las características respecto al TARGET.

Presentamos un cuadro con la evolución de la carga de los distintos ficheros. Considera la carga con todos los valores y sin los Missing Values (MV)



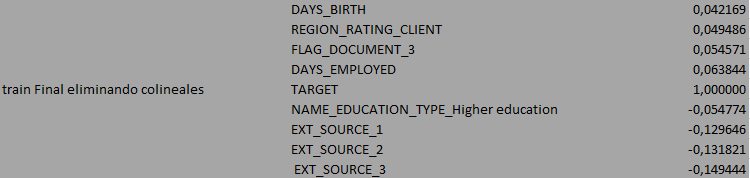
En cuanto a la evolución de la correlación una vez terminada la carga del fichero de entrenamiento (train) realizamos una comparación entre variables también. Considera la carga de los valores sin los Missing Values (MV). En planilla Excel mostramos su evolución completa.



Respecto a esto como mencionamos no vemos una relación fuerte de ninguna de las variables pero si que siempre destaca los campos EXT\_SOURCE (1 – 2 – 3).

Por otro lado, dado que se han generado características extras, también se trató el tema de la colinealidad que podía existir entre las variables. Para ello se aplicó una función y se eliminaron 35 variables.

Analizamos nuevamente el impacto sobre el indicador

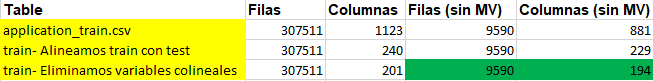


Dicho esto para la aplicación de los modelos decidimos cargar nuevamente el script eliminando los Missing Values (MV) y trabajar con el dataframe train.

De todas formas, alineamos los DF de datos de prueba y capacitación, lo que significa hacer coincidir las columnas para que tengan exactamente las mismas columnas.

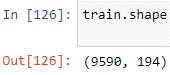
Los marcos de datos ahora tienen las mismas columnas (con la excepción de la columna OBJETIVO en los datos de entrenamiento). Esto significa que podemos usarlos en un modelo de aprendizaje automático que necesita ver las mismas columnas en los marcos de datos de capacitación y prueba.

Mostramos las modificaciones



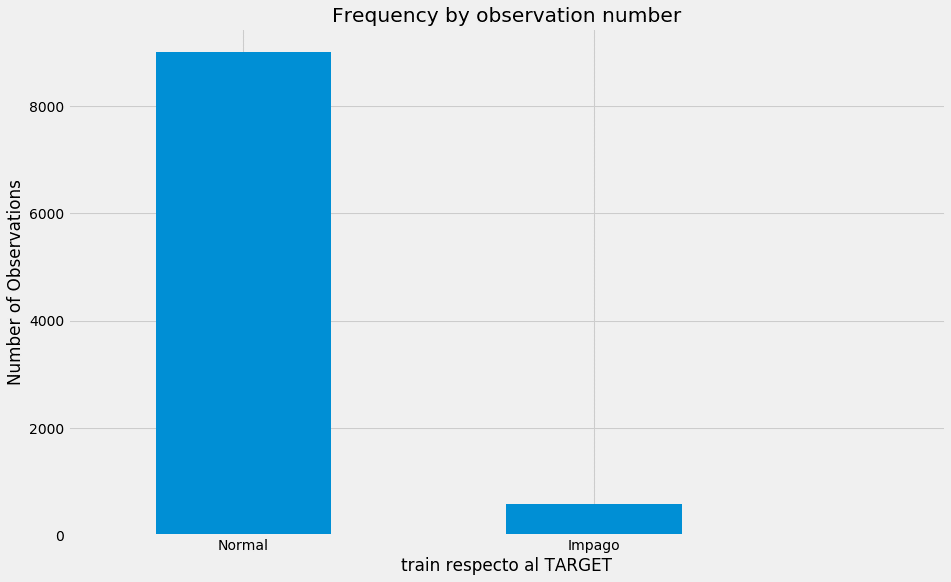
**4. Evaluación y comparación de modelos. Modelos Supervisados**

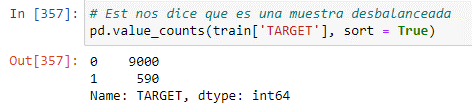
Para comenzar verificamos la composición de nuestro fichero



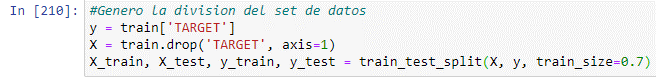
9590 préstamos con 194 características.

Graficamos la distribución del set respecto a la variable objetivo (TARGET).



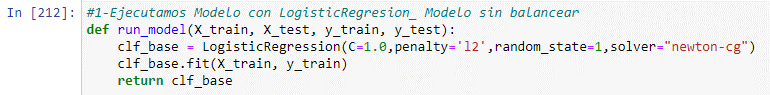


A partir del fichero train definido generamos el split para aplicar a los modelos

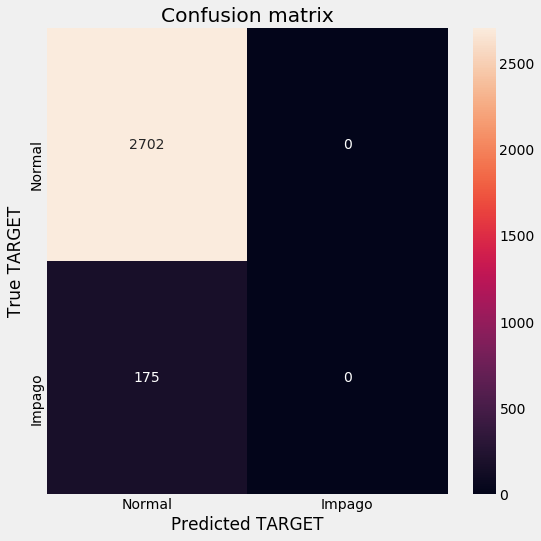


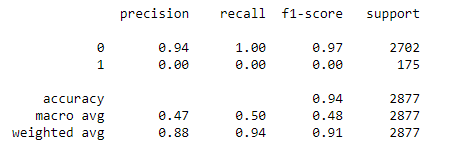
Primero ejecutamos el modelo de LogisticRegresion como referencia

La regresión logística es un algoritmo de clasificación. Se utiliza para estimar valores binarios como 0/1, sí / no, verdadero / falso en función del conjunto dado de variables independientes. Predice la probabilidad de ocurrencia de un evento ajustando datos a una función logit. Como predice la probabilidad, sus valores de salida se encuentran entre 0 y 1



Probamos como responde en el test set y analizamos resultados con una Confusion Matrix





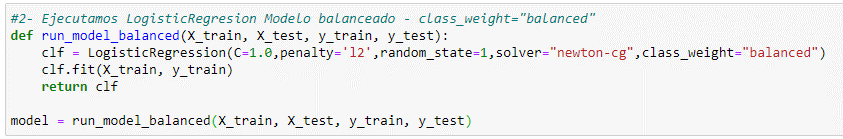
Nuestro modelo no está funcionando. No detecta prestamos que estén impagos (1). Observando la matriz vemos que tenemos cero casos positivos e indicadores de precisión (cercanía del resultado de una medición al valor verdadero) del 0% y un recall (capacidad de discriminar los casos positivos de los negativos) del 0% para los valores que intentamos predecir (1). Esto es resultado del desbalanceo.

En los problemas de clasificación en donde tenemos que etiquetar, (en nuestro caso entre pago (0) e impago (1)), podemos encontrar que nuestro conjunto de datos de entrenamiento contenga alguna característica que sea una clase “minoritaria” es decir, de la cual tenemos muy poquitas muestras. Esto provoca un desbalanceo en los datos que utilizaremos para el entrenamiento de nuestra máquina.

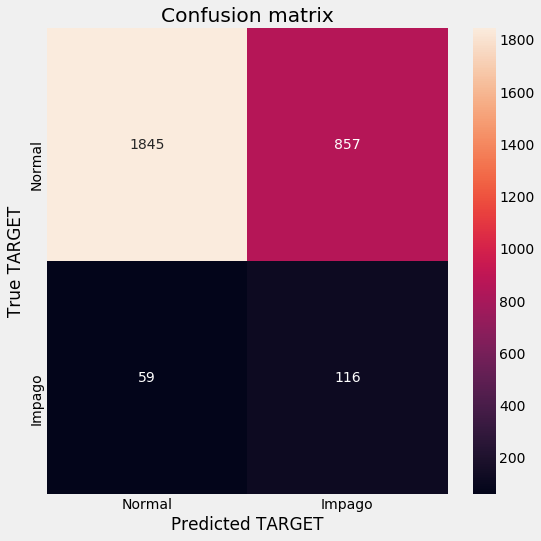
Esto puede suceder en el caso que estamos tratando porque la mayoría de los clientes cancelan sus préstamos quedando pocos casos etiquetados como impagos.

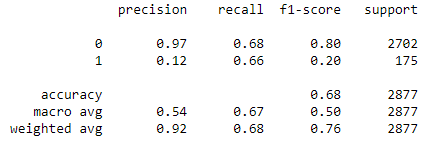
Cuando tenemos un dataset con desequilibrio, suele ocurrir que obtenemos un alto valor de precisión en la clase Mayoritaria y un bajo recall en la clase Minoritaria

Para darle una solución a este inconveniente volvemos a aplicar el modelo con un ajuste por balanceo



Probamos como responde en el test set y analizamos resultados con una Confusion Matrix





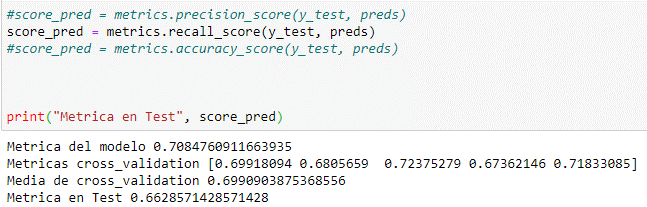
Ahora vemos una mejora. Nuestro modelo, ajustado por balanceo, detecto préstamos impagos acertando en 116 muestras y fallando en 59, dando un recall de 0.66 (True Positive Rate).

En cuanto a nuestro indicador f1- score de 0.20 parecería que hubieran “empeorado” los resultados, se han etiquetado 857 Falsos Positivos cuando no lo eran.

A esto también lo podemos leer como que el modelo esta mejorando en la detección de casos.

Por otro lado y para reforzar nuestra idea de mejora del modelo, realizamos una validación cruzada de nuestra Regresión aplicando KFold con cinco interacciones

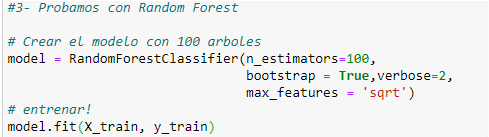




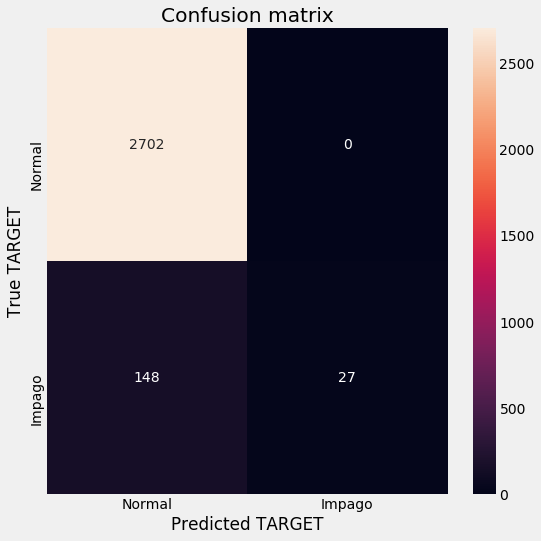
Aquí validamos nuestros indicadores mostrando solo el de Recall.

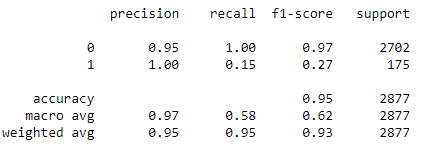
Otro de los modelos que aplicamos fue el de Random Forest

Random Forest es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.



Lo ejecutamos y probamos como responde en el test set y analizamos resultados con una Confusion matrix

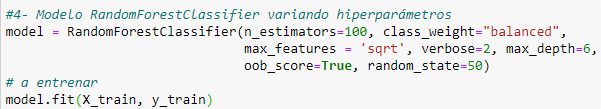


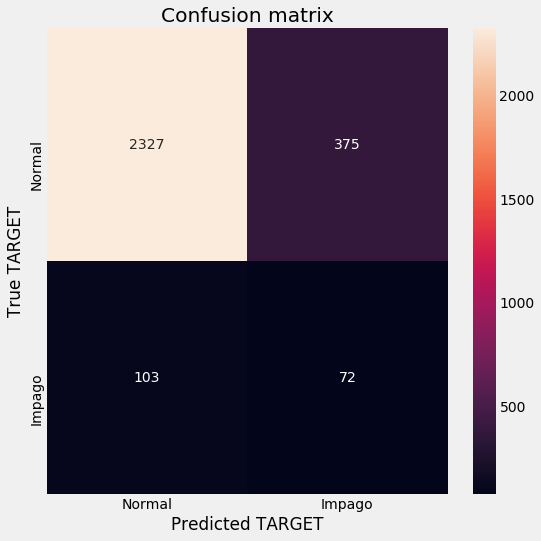


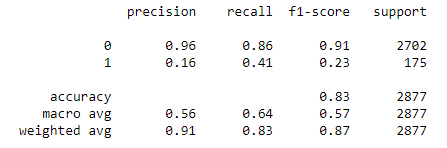
Nuestro modelo ahora nos está mostrando un recall (capacidad de discriminar los casos positivos de los negativos) del 15%. Esta disminución nos dice que el modelo no esta siendo efectivo en el objetivo de detectar que un préstamo no se cobre con las características que definimos. Esto también puede deberse al desbalanceo

Clasificadores de machine learning como Random Forests no lidian muy bien con datasets de entrenamiento desbalanceados ya que son sensibles a las proporciones de las diferentes clases favoreciendo a la clase mayoritaria.

Probemos nuevamente el Modelo RandomForestClassifier variando hiperparámetros. Probamos como responde en el test set y analizamos resultados con una Confusion matrix.







Nuestro modelo ahora nos está mostrando una precisión (cercanía del resultado de una medición al valor verdadero) del 16% y un recall (capacidad de discriminar los casos positivos de los negativos) del 41%.

Si comparamos estos resultados con los del algoritmo de Regresión Logística, vemos que el Random Forest nos dio menos falsos positivos pero la regresión mejores métricas.

En general, la regresión logística funciona mejor cuando el número de variables de ruido es menor o igual que el número de variables explicativas y el bosque aleatorio tiene una tasa positiva verdadera y falsa más alta a medida que aumenta el número de variables explicativas en un conjunto de datos.

Una característica de Random Forest que también nos podría estar afectando es que el modelo no funciona bien con datasets pequeños.

En cuanto al accuracy y aplicable a todos los modelos presentados pierde relevancia cuando los set de datos están desbalanceados. El indicador se calcula básicamente como el número total de predicciones correctas dividido por el número total de predicciones. Excepto en el de regresión ajustada el indicador muestra valores por encima del 80% y puede llevarnos a interpretar que el modelo funciona cuando tal vez no se ha logrado identificar ningún préstamo impago.

**5. Informe final**

Como conclusión podemos decir que el modelo a elegir sería de Logistic Regresion aunque si nuestro objetivo es predecir si un préstamo se pagara o no, con un indicador de recall del 0.66, constituye un buen punto de partida pero hay que mejorarlo.

Desde que comenzamos preferimos trabajar con la mayor cantidad de variables posibles y hasta generamos nuevas a partir de las existentes para ubicar las que mejor predigan nuestro objetivo.

Si bien fuimos relacionando los ficheros históricos de préstamos con nuestro set de datos de entrenamiento y definiendo en última instancia las características con la correlación más alta respecto al campo objetivo nos falta una parte importante que es hablar con algún referente del ente que encargo la tarea para aclarar detalles finos de las características.

Por ejemplo, nosotros seleccionamos los campos EXT\_SOURCE\_1, EXT\_SOURCE\_2, EXT\_SOURCE\_3 por su correlación al objetivo y como información sabemos que solo son puntuaciones normalizadas de fuente de datos externa.

Con la información del cliente podríamos mejorar nuestra selección o combinar información de variables que ayuden a nuestro modelo a un mejor funcionamiento.

Otra cuestión que puede estar afectando a nuestro modelo es la colinealidad (correlaciones entre las mismas variables del fichero). Si bien al set de datos se le aplicó una función para su corrección, por nuestro método de selección de variables pueden haberse descartado alguna(s) importante en cuanto a lo que queremos predecir.