Master in Management + Analytics

Modelos de predicción de scoring crediticio utilizando algoritmos cost-sensitive de machine learning y datos alternativos

Autoría: Gorosabel, Lucas
Fecha de defensa de la tesis: 2023

UNIVERSIDAD TORCUATO DI TELLA

Master in Management + Analytics

Modelos de predicción de scoring crediticio utilizando algoritmos cost-sensitive de machine learning y datos alternativos

Febrero 2023

Alumno: Lucas Gorosabel
Tutor: Nicolás García Aramouni

Abstract

En los años recientes, las instituciones financieras adoptaron técnicas de machine learning para predecir con mayor éxito la probabilidad de repago ante una solicitud de crédito. Si consideramos también la gran cantidad de datos disponibles, de todo tipo, que pueden ser incorporados a los modelos de entrenamiento podríamos esperar un mejor funcionamiento del mercado de créditos. Sin embargo, este no siempre es el caso ya que los sistemas de predicción pueden no obtener una mejora en términos de costos para la institución crediticia ni tampoco una respuesta más satisfactoria para los solicitantes. Este trabajo analiza el impacto de la incorporación de distintas metodologías de datos en la elaboración de modelos de predicción crediticia. Se comparan modelos de aprendizaje automático tradicionales con otros que incorporan costos asimétricos, con el objetivo de identificar los que mejor logran predecir la morosidad. Los resultados muestran la influencia de la elección de algoritmos, modelos de balanceo de clases y tratamiento de variables en las métricas de performance de negocio. Además, se presenta una técnica para minimizar costos que mejora la performance en comparación con modelos que buscan maximizar la métrica accuracy. En este sentido, los resultados indican que la optimización del punto de corte es crucial para minimizar los costos de los modelos de predicción crediticia. A pesar de que los modelos balanceados logran mejores métricas tradicionales, los modelos sin balanceo de clases obtienen menores costos a través de esta técnica. En conclusión, este trabajo destaca la importancia de considerar distintos factores al elegir un modelo de predicción crediticia, con el objetivo de reducir costos y mejorar la satisfacción tanto de los prestamistas como de los solicitantes.

Abstract

In recent years, financial institutions have adopted machine learning techniques to more successfully predict the probability of repayment in response to a loan request. If we also consider the large amount of available data, of all types, that can be incorporated into training models, we could expect better performance from the loan market. However, this is not always the case as prediction systems may not result in cost savings for the lending institution or a more satisfactory response for applicants. This study analyzes the impact of incorporating different data methodologies in the development of credit prediction models. Traditional machine learning models are compared with others that incorporate asymmetric costs, with the goal of identifying those that best predict default. The results show the influence of algorithm choice, class balancing models, and variable treatment on business performance metrics. In addition, a technique for minimizing costs is presented that improves performance compared to models that aim to maximize accuracy. Accordingly, the results indicate that optimizing the threshold is crucial for minimizing the costs of credit prediction models. Despite the fact that balanced models achieve better traditional metrics, models without class balancing achieve lower costs through this technique. In conclusion, this work highlights the importance of considering different factors when choosing a credit prediction model, with the goal of reducing costs and improving satisfaction for both lenders and applicants.

ÍNDICE

Introducción

El acceso al crédito es fundamental para el funcionamiento de la economía, ya que permite a individuos y empresas obtener los recursos financieros necesarios para llevar a cabo sus proyectos. Los bancos, como actores clave en este proceso, utilizan algoritmos de calificación crediticia para determinar la probabilidad de incumplimiento de pago y decidir si aprobar o rechazar un préstamo. En este contexto, el scoring de crédito se ha convertido en una herramienta estándar en la industria financiera para predecir y tomar decisiones automatizadas.

Existen dos limitaciones generales a la hora de diseñar un modelo de scoring crediticio. En primer lugar, un gran desbalance entre los tipos de solicitantes. Por lo general, quienes deben desarrollar modelos de scoring crediticio cuentan con un porcentaje mayoritario de solicitantes que cumplen con sus pagos y un porcentaje minoritario de quienes no cumplen. En segundo lugar, una gran asimetría en el impacto económico entre los escenarios de mala clasificación de solicitudes. Dicha asimetría debe ser considerada en el diseño de los modelos para maximizar la rentabilidad de los prestamistas.

Para superar estas limitaciones, la gran cantidad de datos transaccionales y las nuevas herramientas de procesamiento de datos permiten generar modelos innovadores de scoring crediticio utilizando algoritmos de Machine Learning. Estos modelos incluyen técnicas para corregir el desbalance de clases y así poder predecir en mejor medida a la clase minoritaria en lugar de minimizar la tasa de error. Además de esto, existen algoritmos “cost-sensitive” que adoptan la asimetría de costos ante los distintos escenarios. Estas técnicas mejoran los rendimientos del scoring de crédito modelando mejor el problema del tomador de decisiones a través de una métrica de negocios más adecuada.

Al ver como se suelen resolver este tipo de problemas de credit scoring en foros de ciencia de datos como Kaggle, puede verse que en general se suelen priorizar algoritmos tradicionales por sobre otros que consideran este desbalance de costos. Incorporar este desbalance, de alguna forma, será imperativo para incorporar los potenciales beneficios y costos del negocio en la resolución del problema.

El uso de herramientas de machine learning aplicado a credit scoring no es una novedad. El incremento sustancial en el nivel de cómputo de los últimos años sumado a la gran generación de datos permitieron que muchos autores produzcan material académico relacionado a este problema. Existe literatura que exhibe el comportamiento de los modelos basados en técnicas de machine learning y big data versus los modelos tradicionales (particularmente basados en algoritmos que minan la información lineal entre las variables) a través de

distintas fuentes de datos y tanto en escenarios de estabilidad económica como de shock (Gambacorta et .al, 2019). Otro aspecto de estudio reside esencialmente en comparar las distintas métricas de performance de los modelos de clasificación para credit scoring. En este sentido, hay estudios que muestran cómo los algoritmos de machine learning superan a los modelos tradicionales consistentemente a partir de un gran número de métricas de performance pero a la vez postulan que esta mejora parecería tener un límite y que el foco debería estar en la calidad de los datos utilizados, las métricas de recalibración y en la selección de variables de los modelos (Lessmann et.al, 2015).

En el caso de las bases de datos crediticias es esperable encontrar que los datos se encuentren desbalanceados. Esto hace referencia a que la variable a predecir se compone predominantemente de casos en los que los solicitantes son buenos pagadores y en una pequeña fracción en la que son malos pagadores. Esto presenta un problema para los clasificadores tradicionales que intentan reducir los errores ya que poseen poca información para identificar los casos minoritarios. Existen múltiples técnicas para poder rebalancear de forma sintética los datos y mejorar el poder de predicción sobre los casos minoritarios. SMOTE ("Synthetic Minority Over Sampling Technique) puede ofrecer mejor performance AUC que otros métodos tradicionales en ciertos clasificadores como Naïve Bayes (Chawla et. al, 2002).

A la vez, los modelos predictivos basados en algoritmos simples pueden subestimar el costo total de "misclassification" que se presenta al intentar minimizar los casos de error en las predicciones considerando los tipos de errores (Falsos Positivos y Falsos Negativos) igualmente graves. Ya que los bancos corren con distintos costos en cada uno de estos casos, estas consideraciones deben estar presentes en sus modelos de scoring pero también generan distintas implicancias en la aceptación de solicitudes en relación a cómo fueron implementados. Los clasificadores tradicionales suelen ser insensibles a los costos siendo los mismos implementados para minimizar los errores sin importar el resultado económico resultante. Los algoritmos "cost-sensitive" (sensibles al costo) son una herramienta para lograr introducir esta noción. Existen estudios que muestran que los modelos "cost sensitive" considerando una matriz de costos fijos (costos constantes para todas las observaciones) logran reducir los costos asociados a expensas de aumentar el error de clasificación (Saidi et.al, 2018).

Más aún, se han realizado contribuciones que muestran la importancia que tienen los modelos "cost example dependent" (sensibles al costo por observación) en lograr reducir los costos a diferencia de los modelos que asumen costos fijos para todas las observaciones (Correa Bahnsen, Aouada, Ottersten, 2014). Esto es esperable ya que en el caso del scoring crediticio

asumir que todos los solicitantes presentan la misma matriz de costos se asemeja en pocos sentidos al caso real y más a una simplificación teórica. En la práctica los costos que presentan los solicitantes se relacionan a variables como la tasa de interés, el volumen del préstamo y la modalidad de repago, entre otros, siendo estos particulares en cada contrato. Esta metodología "cost example dependent" se refiere a una matriz de costos que se adapta a cada solicitante.

Muchos de estos trabajos asumen que el costo de clasificar correctamente a un buen pagador es nulo e intentan analizar específicamente el comportamiento sobre los errores de clasificación. Una posibilidad es considerar que ese costo no sea imputado como nulo sino más bien como un costo negativo, haciendo referencia al beneficio que percibe una institución crediticia a través del interés sobre el monto prestado. Este supuesto podría tener grandes implicancias sobre los clasificadores ya que genera mayores incentivos a aceptar solicitudes y se asemeja aún más al caso real de estas compañías.

Los algoritmos "cost-sensitive" pueden ser divididos en dos grupos: Unos se los considera métodos directos en los que el clasificador adquiere los costos en su función objetivo. A los otros se los considera métodos indirectos en los que se convierte un clasificador tradicional insensible al costo a través de distintas técnicas. Una de las técnicas más efectivas para convertir un clasificador en "cost-sensitive" es alterar el punto de corte. Este proceso consta de tomar las predicciones de un clasificador y buscar el punto de corte que minimice los costos. El punto de corte en este caso es la probabilidad que determina si la decisión es aprobar o rechazar un crédito (el punto de corte por default en los clasificadores tradicionales es de 0,5 pero este podría establecerse en otro valor haciendo más o menos flexible la decisión de aprobar o rechazar). Esta metodología resulta mejor que otros procesos para convertir clasificadores en "cost-sensitive" e incluso que los métodos directos (Sheng, Ling, 2006). Por otro lado, es posible imputar costos negativos (en referencia al costo de predecir correctamente a un buen pagador) en esta metodología a diferencia de ciertos métodos directos como los que provee la librería MLR de R Studio. En este caso no solo se puede obtener una aproximación real del caso de estudio sino que también se puede analizar el uso de distintos métodos de balanceo de clases en el contexto de minimizar los costos asociados al negocio.

El presente trabajo tiene como objetivo brindar a los tomadores de decisión en el ámbito crediticio un marco comparativo que muestre el impacto de distintas técnicas de machine learning tanto para minimizar los errores predictivos como también para reducir los costos asociados al negocio de manera práctica. En concreto, utilizando dos muestras de datos distintas extraídas de competencias Kaggle (datos públicos publicados por empresas privadas) se evaluarán un

conjunto de técnicas para tratamiento y selección de variables (principalmente métodos de rebalanceo de clase objetivo), aplicando modelos de clasificación tradicionales como también ensamblados medidos por distintas métricas de performance. Adicionalmente se introducirán distintas estimaciones de costos de clasificación y se implementarán modelos que tengan como objetivo minimizar los costos a través de estructuras con costos fijos y variables por observación exhibiendo el impacto de estos modelos sobre las métricas originales. Finalmente, se presentarán modelos que optimicen el punto de corte de la clasificación logrando minimizar el costo promedio por observación y reflejar la asimetría de los distintos escenarios a través de una extensión de la librería MLR.

La expectativa de este trabajo se basa en poder generar una solución automatizada de scoring crediticio que logre predecir eficientemente la probabilidad de que los aplicantes tengan dificultades para cancelar sus obligaciones. A su vez, se desea obtener evidencia por parte de los modelos cost-sensitive del efecto que tienen las instituciones sobre las aplicaciones de los individuos. De esta forma, se podrán cuantificar en términos estadísticos y económicos el efecto de utilizar modelos de scoring más complejos, las decisiones sobre la parametrización de los modelos y la incorporación de distintas variables en ambos tipos de clientes, lo cual ayuda además a comparar el potencial beneficio que puede llegar a tener este tipo de técnicas para distintos tipos de empresas. Se espera que, si los modelos tienen éxito, los tomadores de decisiones tengan en cuenta estos resultados, no solo en el uso de herramientas de Machine Learning, sino también en la incorporación de técnicas cost-sensitive para lograr una mayor interpretación de los impactos en el negocio.

Se pretende contribuir a la literatura a lo largo de este trabajo abordando las siguientes preguntas:

- ¿Cuál es el impacto del proceso de tratamiento y selección de variables en las principales métricas de performance?
- ¿Cómo es la performance de los modelos tradicionales versus modelos ensamblados?
- ¿Cuál es el impacto en costos de los modelos “cost-sensitive” que asumen costos fijos versus modelos que asumen costos variables por observación?
- ¿Cómo se comportan las técnicas de rebalanceo de clases para los clasificadores que buscan reducir costos?
- ¿Cuál es el trade-off entre optimizar el punto de corte para minimizar los errores de predicción versus minimizar los costos?

Este documento se estructura de la siguiente manera: en la sección 2 se presentarán los datos y métodos. En la sección 3 se exhibirán los modelos. En la sección 4 se mostrarán los resultados obtenidos y el análisis de los mismos. Por último, en la sección 5 se presentarán las conclusiones del trabajo y las derivaciones potenciales del mismo.

Datos y métodos

DATOS

Los datos utilizados en este trabajo fueron extraídos en su totalidad de la plataforma Kaggle. Esta plataforma ofrece competencias sponsoreadas por empresas privadas que brindan bases de datos para que la comunidad de la plataforma pueda experimentar con los datos armando y publicando modelos en búsqueda de optimizar métricas determinadas. De esta forma, las empresas que sponsorean estas competencias acceden a innovación abierta permitiendo que los miles de usuarios de Kaggle puedan desarrollar modelos de machine learning para mejorar sus métricas de negocio.

En este caso, se utilizan datos publicados por dos competencias distintas por un par de instituciones crediticias donde el objetivo es desarrollar modelos que logren predecir con éxito una variable que indique si el solicitante tiene o no problemas de repago maximizando la métrica de performance AUC. La determinación en el uso de más de una serie de datos se basa en poder brindar mayor robustez en el uso de distintas metodologías ya que si los resultados de aplicar una metodología son consistentes en ambas fuentes de datos, estos tendrán mayor validez que si fuera en una sola serie donde los resultados podrían ser específicos para los datos en los que fueron aplicados. Adicionalmente, dado que la comunidad de Kaggle comparte una gran cantidad de información relacionada a las competencias, más precisamente, en muchos casos los equipos ganadores de las competencias expresan los desafíos más relevantes que han tenido que sobrellevar para obtener la mejor métrica e incluso comparten el código para replicar esta performance, es importante destacar que la contribución de este trabajo no es la optimización de la métrica AUC sino un análisis comparativo de las metodologías y particularmente el impacto en costos del negocio.

Dado que este trabajo utiliza más de una base de datos y que el foco se encuentra en encontrar soluciones para el mercado de créditos y no en las capacidades de una institución financiera en particular se realizará un estudio acotado de exploración de los datos. Sin embargo, si se hará hincapié en las características fundamentales sobre la variable objetivo a predecir en cada caso y en las variables que se tomarán en cuenta para estimar los costos de clasificación.

El primer set de datos fue extraído de la competencia “Give me some credit” (GMSC). Esta competencia provee dos archivos descargables de la página de Kaggle: Un dataset para entrenar que incluye la variable target (“cs-training”) y otro dataset para testear y predecir la variable target ante su ausencia (“cs-test”). Dado que el desempeño de los modelos aplicados a los datos de

testing en el marco de la competencia no forma parte del objetivo del presente estudio, el análisis se centra únicamente en el uso del set de datos de entrenamiento. Esta serie de datos incluye 150.000 observaciones y 11 variables detalladas en la siguiente tabla:

Tabla 1: Variables GMSC

Nombre de variable	Descripción	Tipo
SeriousDlqin2yrs (TARGET)	**Persona con 90 días de morosidad o más.**	**Binario (0/1)**
RevolvingUtilizationOfUnsecuredLines	Saldo total de tarjetas de crédito y líneas de crédito personales, excepto bienes inmuebles y ninguna deuda a plazos, como préstamos para automóviles, dividido por la suma de los límites de crédito.	Porcentual
Age	Edad del solicitante.	Entero
NumberOfTime30-59DaysPastDueNotWorse	Número de veces que el solicitante ha estado atrasado entre 30 y 59 días, pero no peor en los últimos 2 años.	Entero
DebtRatio	Pagos mensuales de deuda, pensión alimenticia, costos de vida divididos por el ingreso bruto mensual	Porcentual
MonthlyIncome	Ingreso Mensual.	Real
NumberOfOpenCreditLinesAndLoans	Número de préstamos abiertos (cuotas como préstamos para automóviles o hipotecas) y líneas de crédito (por ejemplo, tarjetas de crédito).	Entero
NumberOfTimes90DaysLate	Número de veces que el solicitante ha	Entero

	estado atrasado 90 días o más.	
NumberRealEstateLoansOrLines	Número de préstamos hipotecarios e inmobiliarios, incluidas líneas de crédito sobre el valor neto de la vivienda.	Entero
NumberOfTime60-89DaysPastDueNotWorse	Número de veces que el solicitante ha vencido 60-89 días, pero no ha empeorado en los últimos 2 años.	Entero
NumberOfDependents	Número de dependientes en su familia excluyéndose a sí mismos (cónyuge, hijos, etc.).	Entero

El objetivo en el caso de esta competencia es predecir el valor de la variable *SeriousDlqin2yrs*. Esta variable indica si una persona experimenta 90 días o más de morosidad desde que ha obtenido un préstamo. Se puede considerar que un solicitante podría encontrarse apto de repagar su préstamo incluso con 90 días. Sin embargo, dado que la variable es fijada por una institución crediticia consideramos que es suficiente para pensar que un solicitante no es apto de recibir el préstamo. Si el valor de esta variable es cero, el solicitante no tuvo inconvenientes para cancelar sus obligaciones contractuales en un tiempo menor a 90 días. Si el valor de esta variable es uno, el caso es el contrario.

Este set de datos presenta 10 variables independientes sin presencia de variables categóricas (variables con un número limitado de valores o categorías). Estas variables consisten en datos fundamentales para las instituciones crediticias a la hora de elaborar modelos predictivos y podrían ser consideradas dentro de los modelos tradicionales. En concreto brindan información sobre la liquidez del solicitante (*MonthlyIncome*), su historial crediticio (*NumberOfTimes90DaysLate*) y características generales (*Age* / *NumberOfDependents*).

La presencia de valores ausentes (missings/NA) es importante de destacar. Se podría atribuir la presencia de estos valores a una evasión por parte del solicitante a compartir el dato en cuestión como también a la institución crediticia que imputa un valor ausente para volver más complejo el análisis intencionalmente. Más allá del motivo por el cual esto se presenta, es común

encontrar estos valores en bases de datos y consecuentemente deben ser tratados bajo alguna metodología en particular como puede ser eliminarlos (por observación o variables), reemplazarlos por valores promedios (o por la clase mayoritaria para las variables categóricas o binarias), entre otras estrategias. En el presente trabajo se aplican varias estrategias para tratar estos datos y encontrar la que mejor optimiza los modelos. Para el caso de este set de datos, la variable *MonthlyIncome* presenta 29.731 observaciones con valores ausentes y *NumberOfDependents* presenta 3.924 observaciones con valores ausentes. Este set de datos presenta 29.731 observaciones con datos ausentes sobre las 150.000 observaciones, es decir, 19,82% de las observaciones presentan valores ausentes.

La característica fundamental en este conjunto de datos es el desbalance de la clase target *SeriousDlqin2yrs.* El mismo muestra que la variable toma valor cero (repaga su préstamo) en un 93,3% de los casos, y toma valor uno (tiene inconvenientes para pagarlo) en un 6,7% de los casos. Esto significa que el modelo a diseñar tendría una gran asimetría de casos para intentar predecir la clase positiva. En este sentido, existen metodologías que permiten equilibrar esta relación entre ambas clases reduciendo casos o creando nuevos. Varios métodos son explicados, utilizados y comparados que contribuyen al análisis mostrando la eficiencia de cada uno de ellos.

El segundo set de datos fue extraído de la competencia de Kaggle “Home Credit Default Risk”. Esta competencia presenta una base de datos de entrenamiento primaria, 6 bases de datos de entrenamiento relacionales con información específica (historial crediticio, balance de tarjeta de crédito, entre otros) y una base de datos de testeo. En este caso, se utilizará únicamente la base de datos de entrenamiento primaria dado que dicha base contiene 307.511 observaciones (solicitantes) y 122 variables, es decir, suficientemente mayor que el de la competencia GMSC. De esta manera, el trabajo abarca dos set de datos con distinta dimensionalidad y tipo de información.

La competencia premia a quien alcance la máxima puntuación de la métrica AUC al predecir la variable *TARGET*. Esta variable toma valor uno en caso de que el cliente presente dificultades de repago y valor cero en el caso contrario. Al igual que en la competencia GMSC, esta variable está altamente desbalanceada ya que en el 91,9% de los casos se trata de casos en los que el solicitante no tiene dificultades de repago y solamente un 8,1% de los casos si las tiene. Por lo tanto, los métodos de rebalanceo serán considerados para abordar esta problemática en ambos set de datos y analizados comparativamente.

Las 122 variables contenidas en los datos de entrenamiento incluyen información acerca de la situación financiera del solicitante (ej. Ingreso),

scoring de bases externas, tipo de préstamo, situación (ej. hora de procesamiento de la solicitud), datos de la ubicación donde reside, entre otros (la descripción de cada una de ellas se encuentra presente en Anexo 1- Descripción de variables de la base de datos HCDR). Dado que se trata de una gran cantidad de información reflejada en una alta cantidad de variables, es importante considerar que muchas de estas podrían tener poco poder explicativo al ser incluidas en los modelos de entrenamiento. A su vez, considerar una alta dimensionalidad en los modelos de entrenamiento demanda un alto nivel de computo. Es por eso que en la elaboración de modelos se considerarán distintos métodos de selección de variables. Un método que no se considerará es el “Principal Component Analysis” (PCA) dado que mediante su aplicación se pierde el poder explicativo de las variables.

Otra característica relevante de este set de datos es la incorporación de variables categóricas. Estas variables presentan inconvenientes al ser aplicadas a ciertos métodos como los algoritmos regresivos, por lo que para ser incluidas deben ser transformadas en variables numéricas. Lo mismo sucede con los valores ausentes anteriormente mencionados. En este caso se presentan en 61 de las 122 variables, es decir, la mitad de las variables contienen esta particularidad, mientras que la cantidad de observaciones sin valores ausentes es de 11.351 (3,7%). Dada la gran presencia de datos ausentes es evidente que eliminar la mitad de las variables o el 96,3% de las observaciones podría significar una gran pérdida de información por lo que es muy relevante el tratamiento de estos valores. Se aplicarán varios métodos para resolver esta problemática.

EXPLORACIÓN DE DATOS

A continuación se analizan las relaciones entre las variables explicativas más relevantes y la variable objetivo para cada base de datos:

Figura 1: Conteo de observaciones por rango etario y clase, GMSC

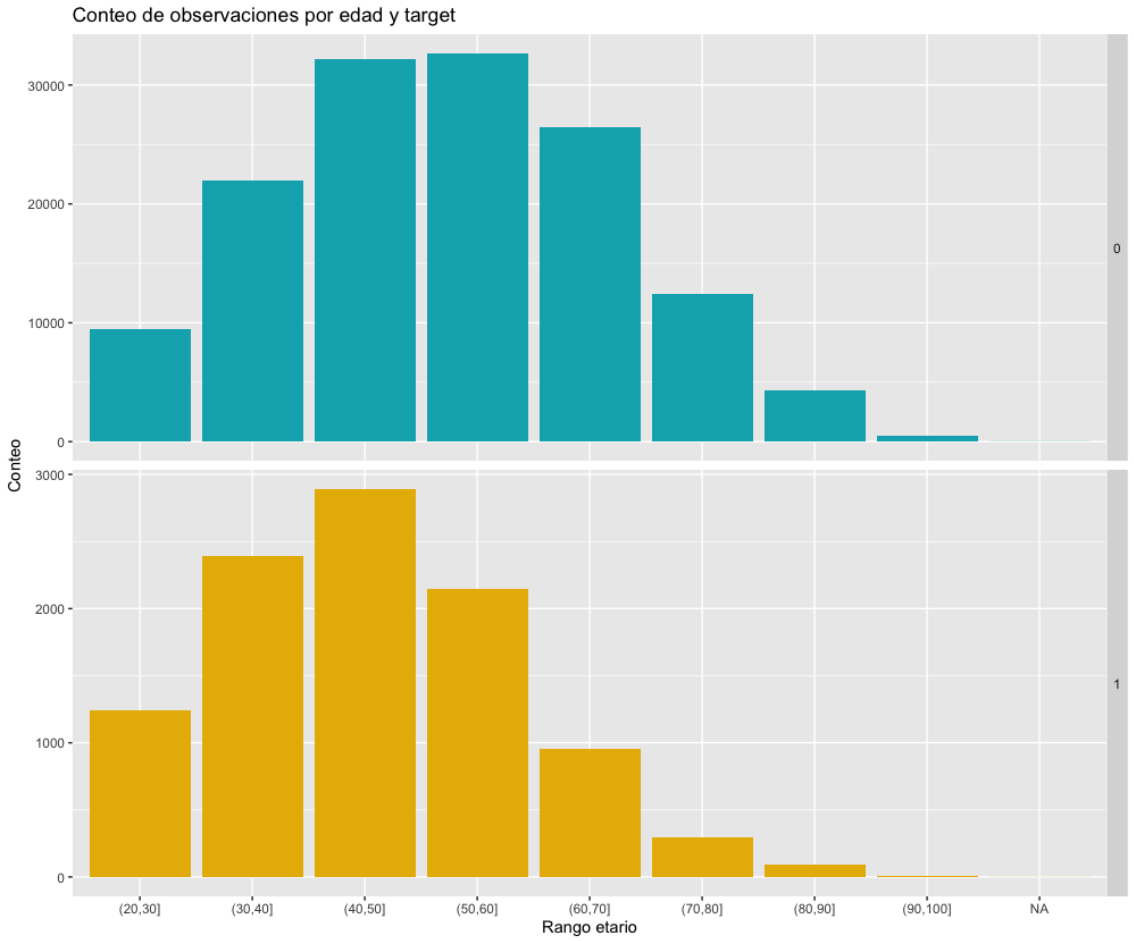
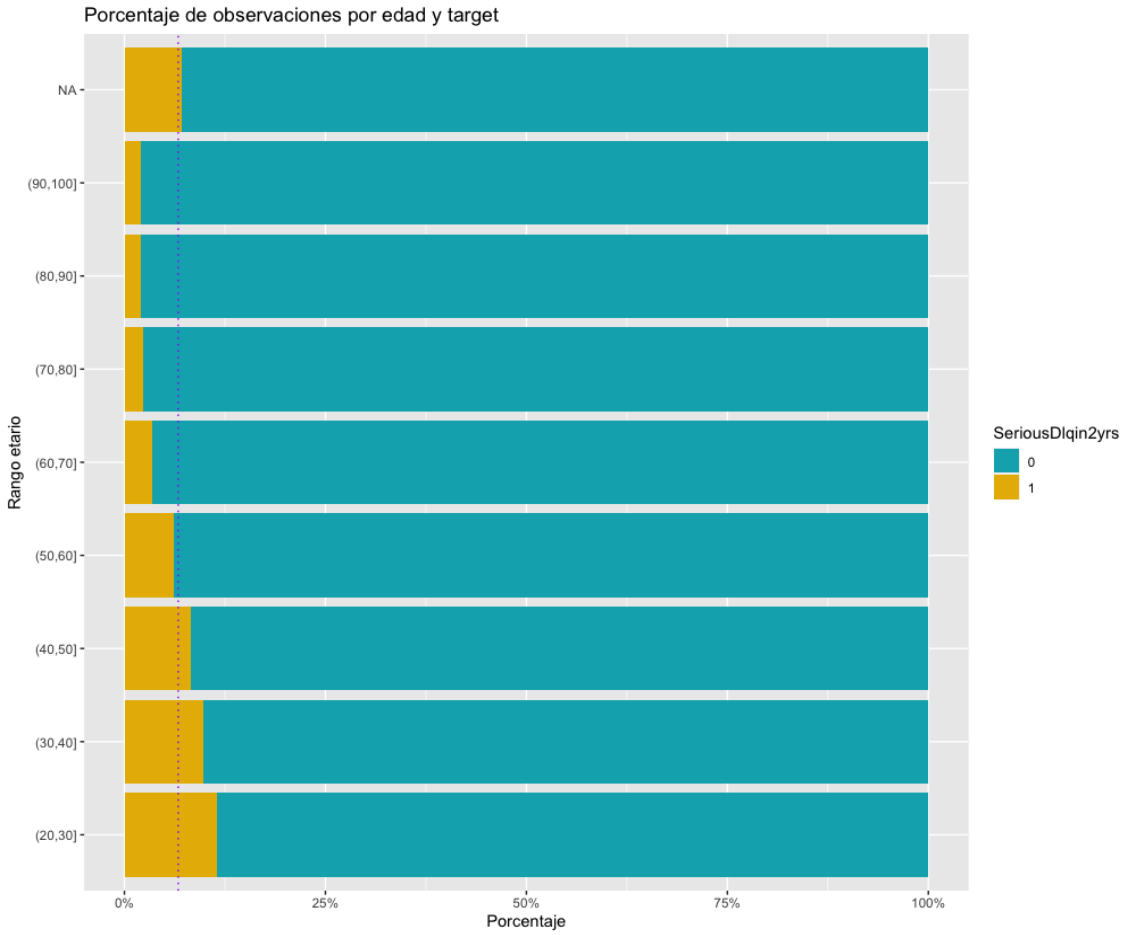


Figura 2: Porcentaje de observaciones por edad y clase, GMSC



La figura 2 muestra el porcentaje de observaciones de cada clase para cada rango etario. La línea punteada exhibe el porcentaje de observaciones totales pertenecientes a la clase minoritaria (6,7%). Lo que se puede observar es que el porcentaje de clase minoritaria varía en relación al rango etario. Más aún, el gráfico muestra que la probabilidad de que el solicitante presente problemas de repago de crédito es mayor para los rangos etarios más bajos. Una posible explicación es que a mayor edad se presume mayores ingresos y una mayor solvencia crediticia.

Figura 3: Conteo de observaciones por ingreso mensual y clase, GMSC

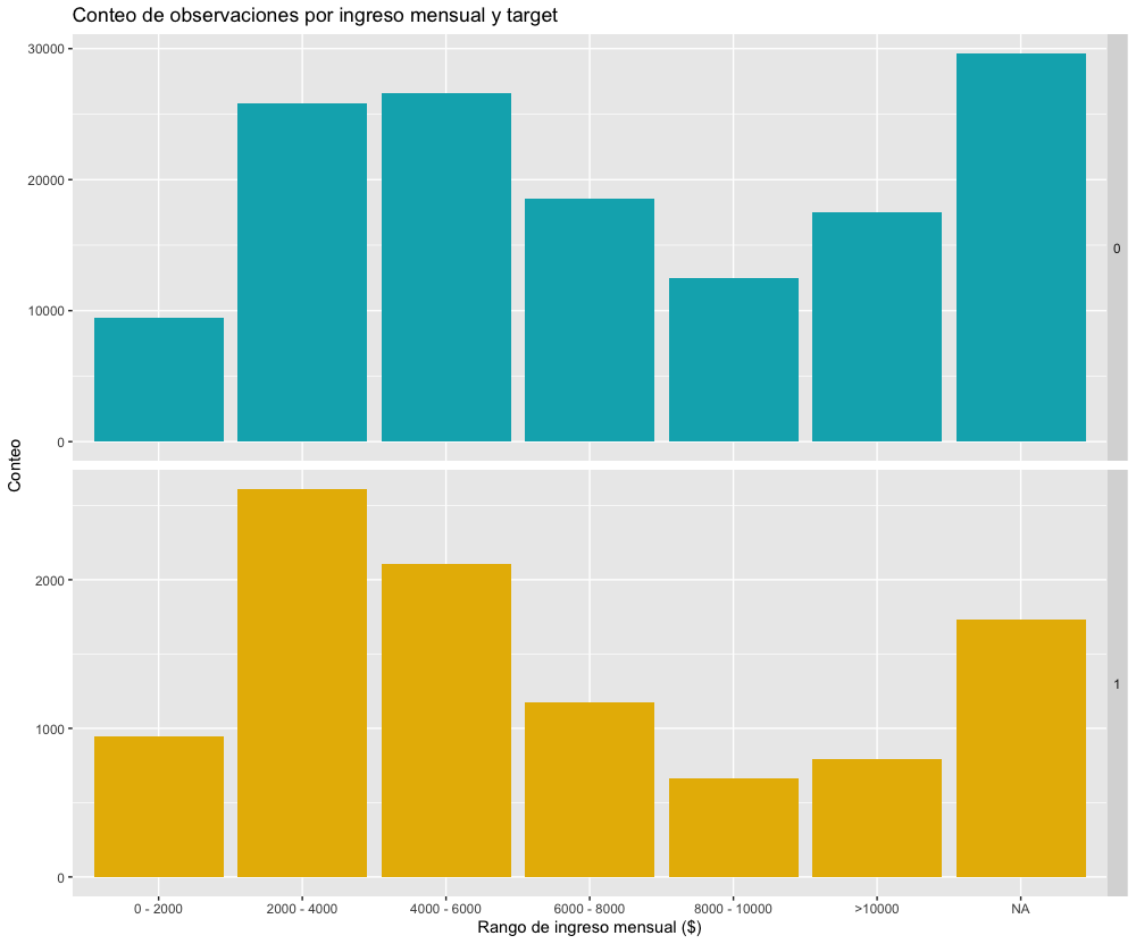
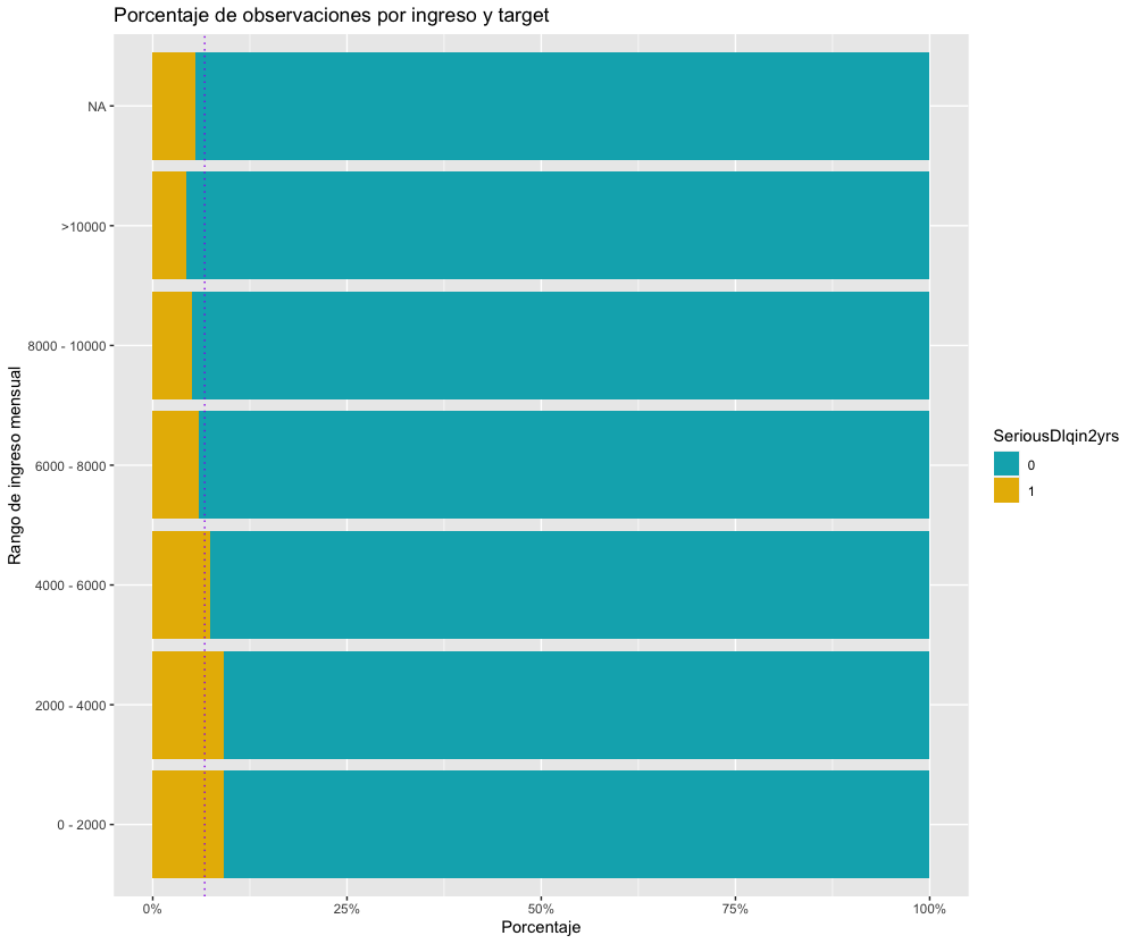


Figura 4: Porcentaje de observaciones por ingreso mensual y clase, GMSC



La figura 4 muestra el porcentaje de observaciones de cada clase para cada rango de ingreso mensual. Se observa que los rangos de menores ingresos tienen un porcentaje mayor de clase minoritaria. A medida que el ingreso aumenta, el porcentaje de clase minoritaria disminuye. Al igual que con la edad, el ingreso muestra una tendencia que indica que a mayor ingreso la probabilidad de que un solicitante tenga problemas de repago es menor.

Figura 5: Conteo de observaciones por ratio de deuda y clase, GMSC

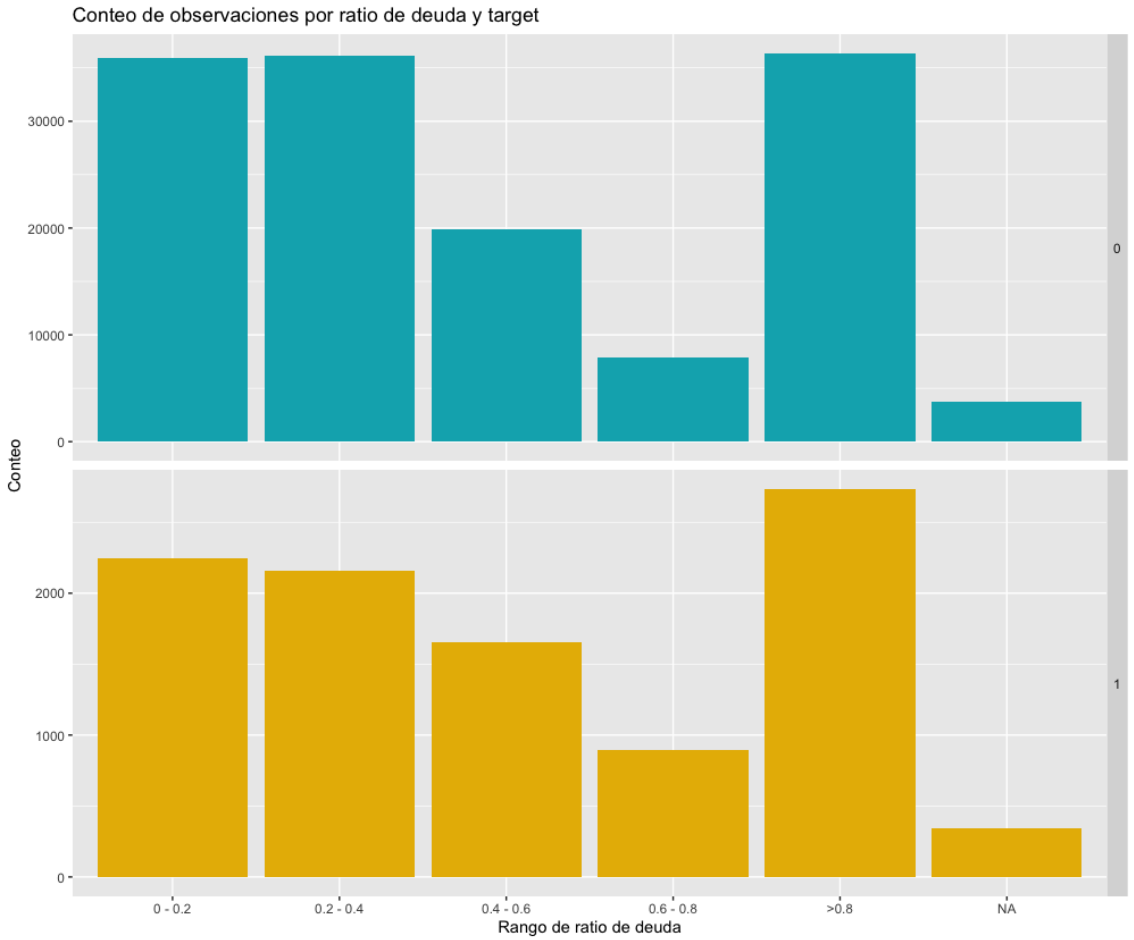
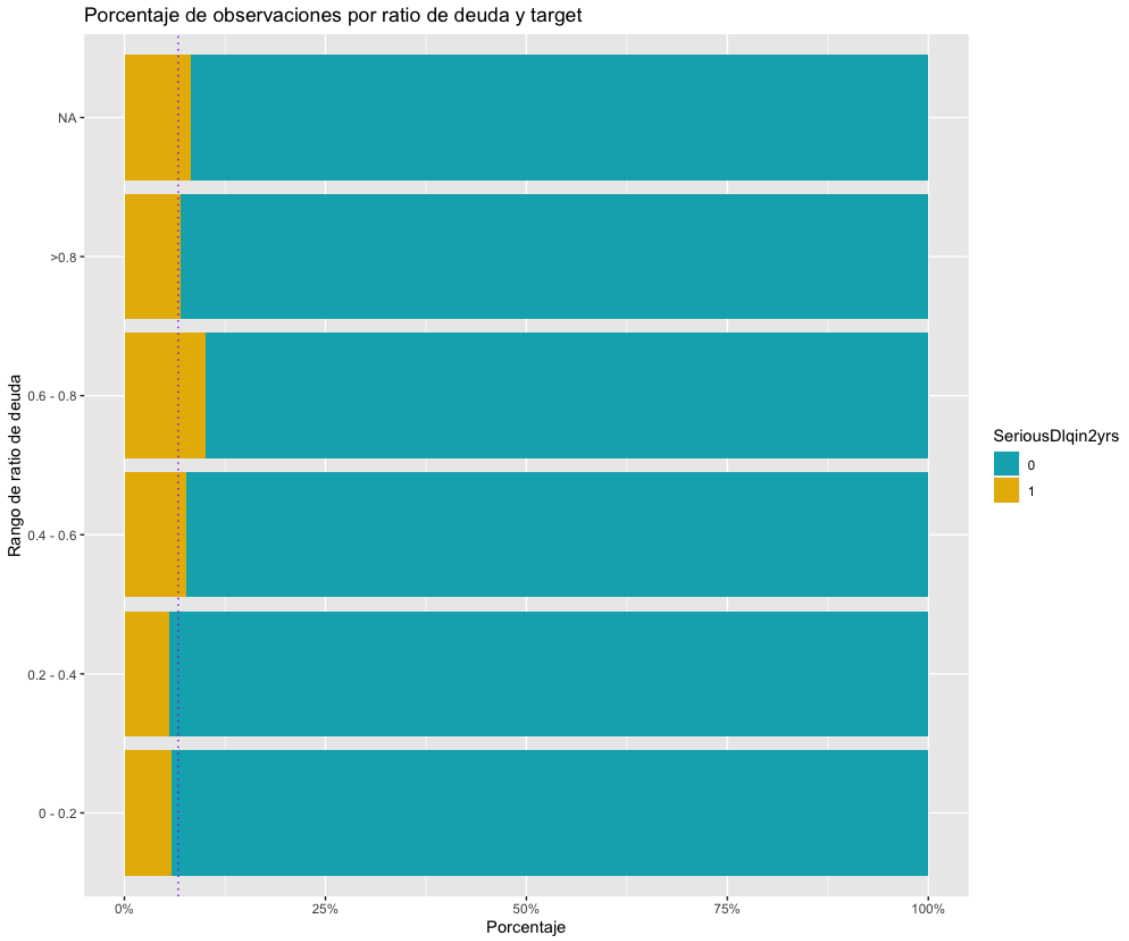
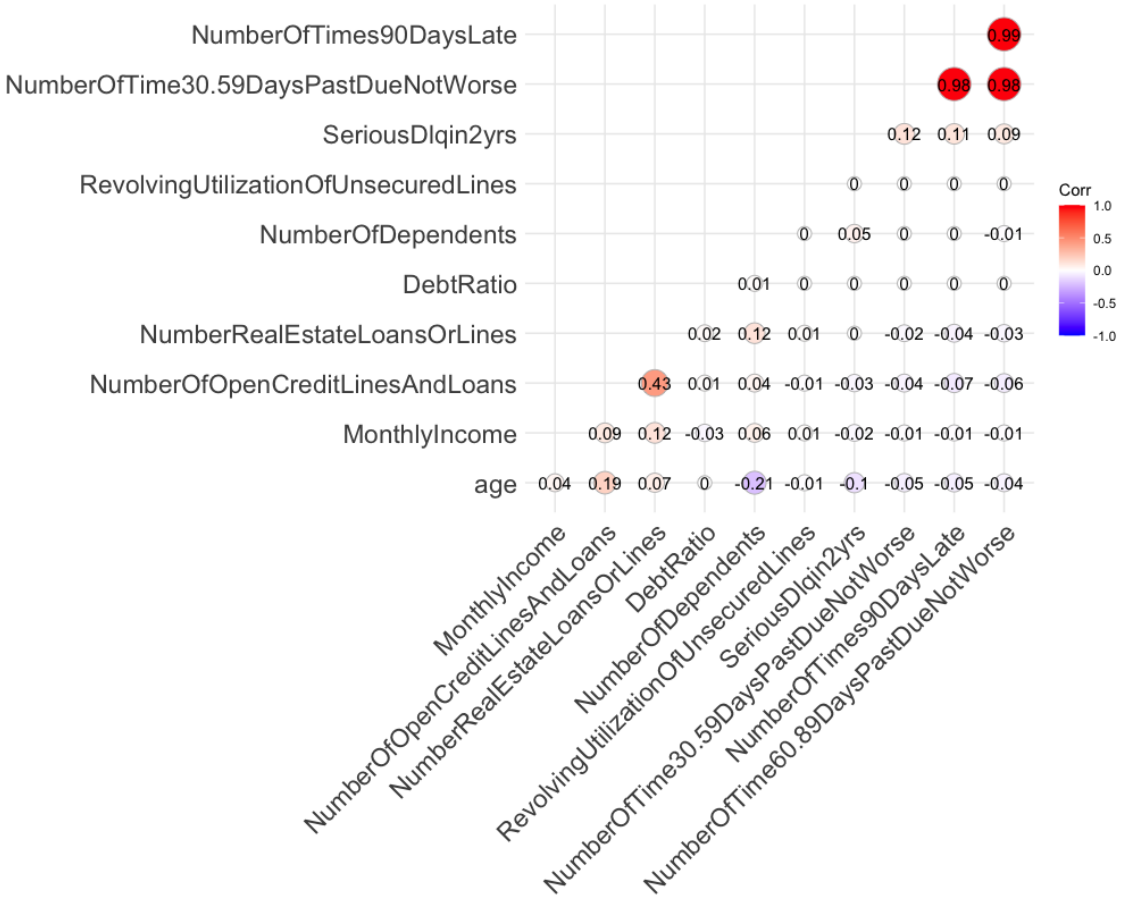


Figura 6: Porcentaje de observaciones por ratio de deuda y clase, GMSC



La figura 6 muestra el porcentaje de observaciones de cada clase para cada rango de ratio de deuda (*DebtRatio*). En este caso se observa que en el rango 0,4 a 0,8 la cantidad de solicitantes de la clase minoritaria es mayor al promedio. En el rango 0 a 0,4 la cantidad de solicitantes de la clase minoritaria es menor al promedio mientras que para valores mayores a 0,8 la proporción se encuentra cercano al promedio (0,67). Es importante destacar que 28.877 observaciones sobre las 150.000 observaciones totales presentan valores de *DebtRatio* mayores a 10, es decir, casi un 20% de las observaciones presentan pagos de deuda sobre ingresos mensuales muy altos a pesar de que en este rango no se visualiza una diferencia de proporciones entre las clases significativa.

Figura 7: Matriz de correlaciones, GMSC



La figura 7 muestra la matriz de correlaciones entre las variables de la base de datos GMSC. Lo que se puede observar es que hay 3 pares de variables con una alta correlación de más de 0.98 (los 3 pares resultantes de las interacciones entre las variables "*NumberOfTimes*.."). Estas variables hacen referencia a la cantidad de veces que un solicitante estuvo demorado en sus pagos de deuda en los últimos 2 años por "x" cantidad de días, por lo que es esperable que exista una cierta correlación entre estas variables (si el solicitante tiene más de 90 días de retraso también tuvo 30 días de retraso), aunque el gráfico muestra una correlación positiva casi perfecta. Esto es relevante ya que a la hora de entrenar los modelos sería necesario eliminar 2 de las 3 variables o realizar una transformación que contemple la información de las 3 variables involucradas (por ejemplo la suma de las 3 variables o el promedio).

En cuanto al resto de los pares de variables, todos los valores presentan bajas correlaciones de manera que todas podrían ser incluidas en el entrenamiento de los modelos.

Figura 8: Conteo de observaciones por ingreso anual y clase, HCDR

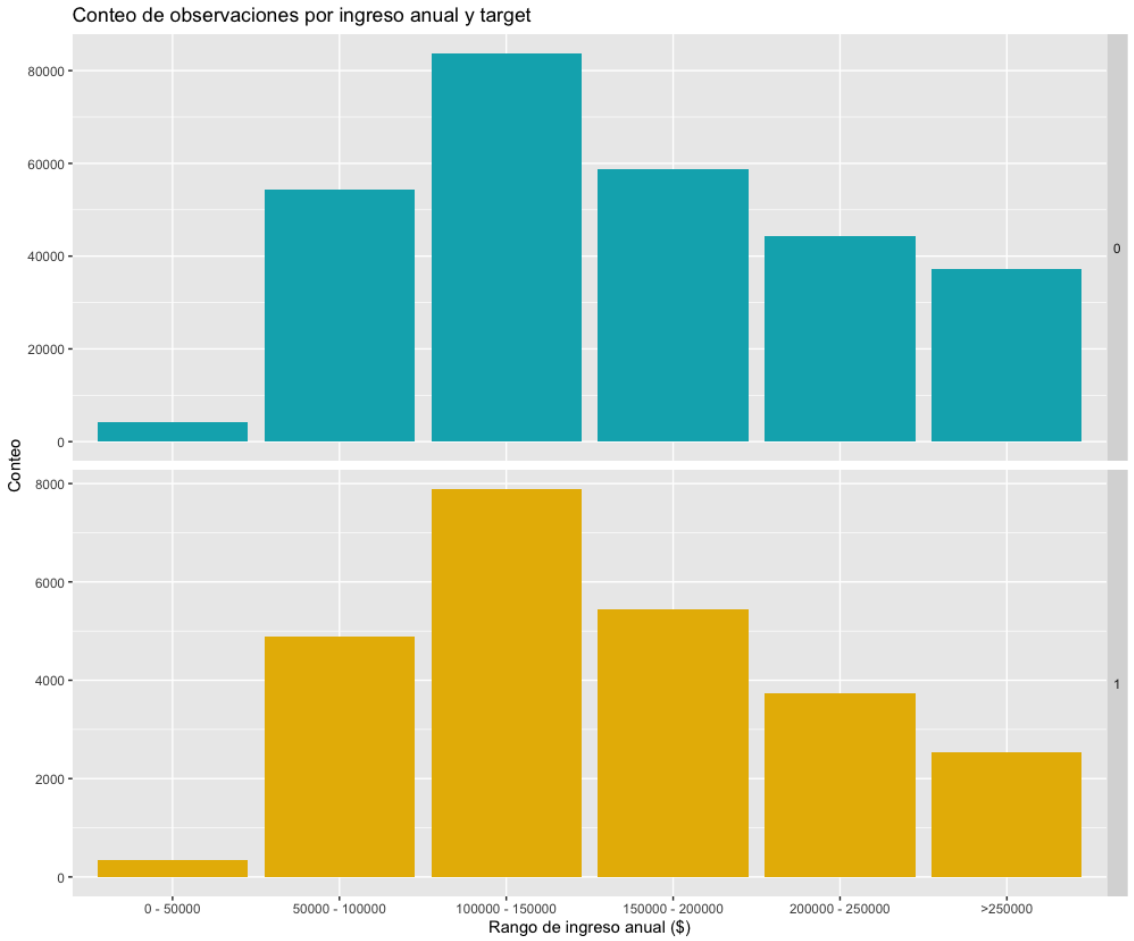
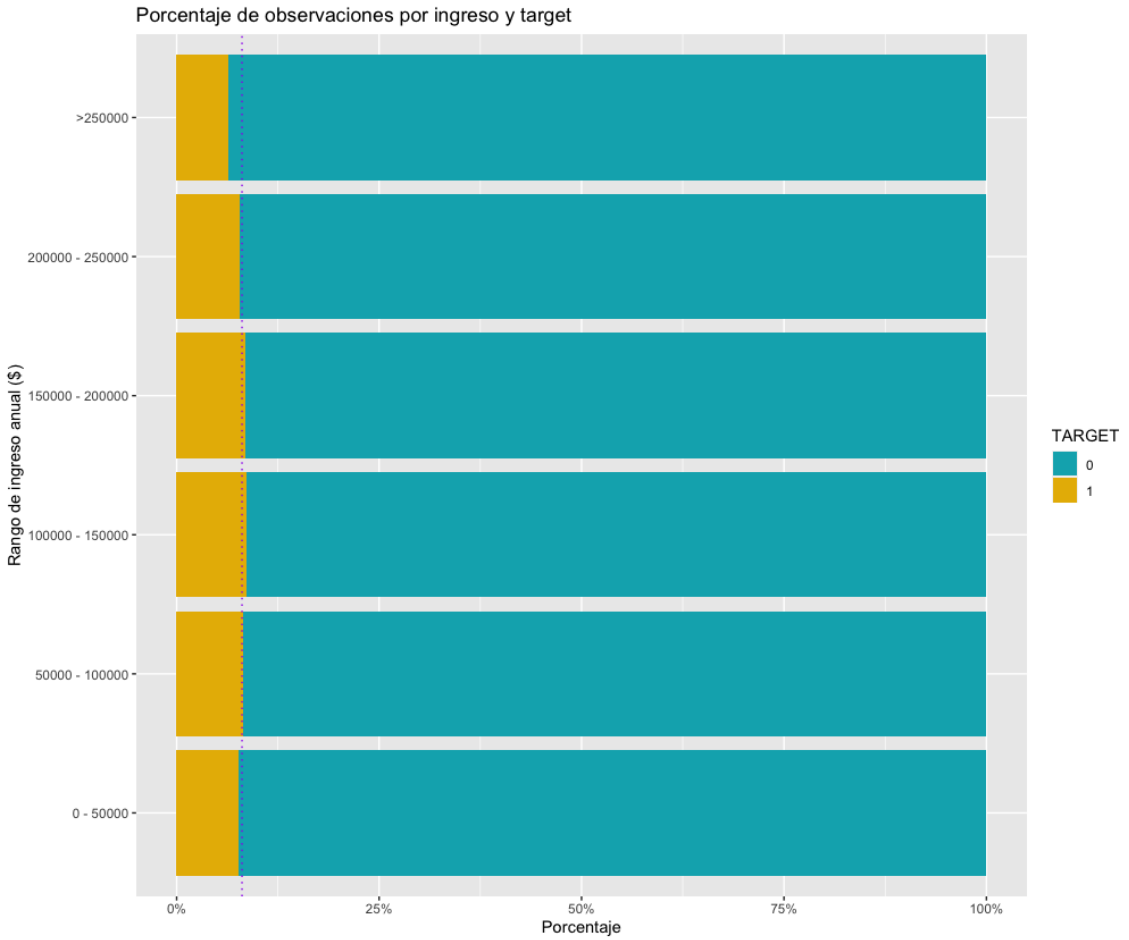


Figura 9: Porcentaje de observaciones por ingreso anual y clase, HCDR



La figura 9 muestra el porcentaje de observaciones de cada clase para cada rango de ingreso anual. La línea punteada exhibe el porcentaje de observaciones totales pertenecientes a la clase minoritaria (8%). Se observa que el porcentaje de clase minoritaria se encuentra cercano al promedio general (8%) para cada rango de ingreso. La diferencia más significativa se observa en el rango de mayor ingreso (>$250.000) para el cual la clase minoritaria tiene una proporción menor al promedio. Este resultado es esperable considerando que un mayor salario presenta mejores condiciones de repago sin considerar el monto del crédito solicitado.

Figura 10: Conteo de observaciones por crédito y clase, HCDR

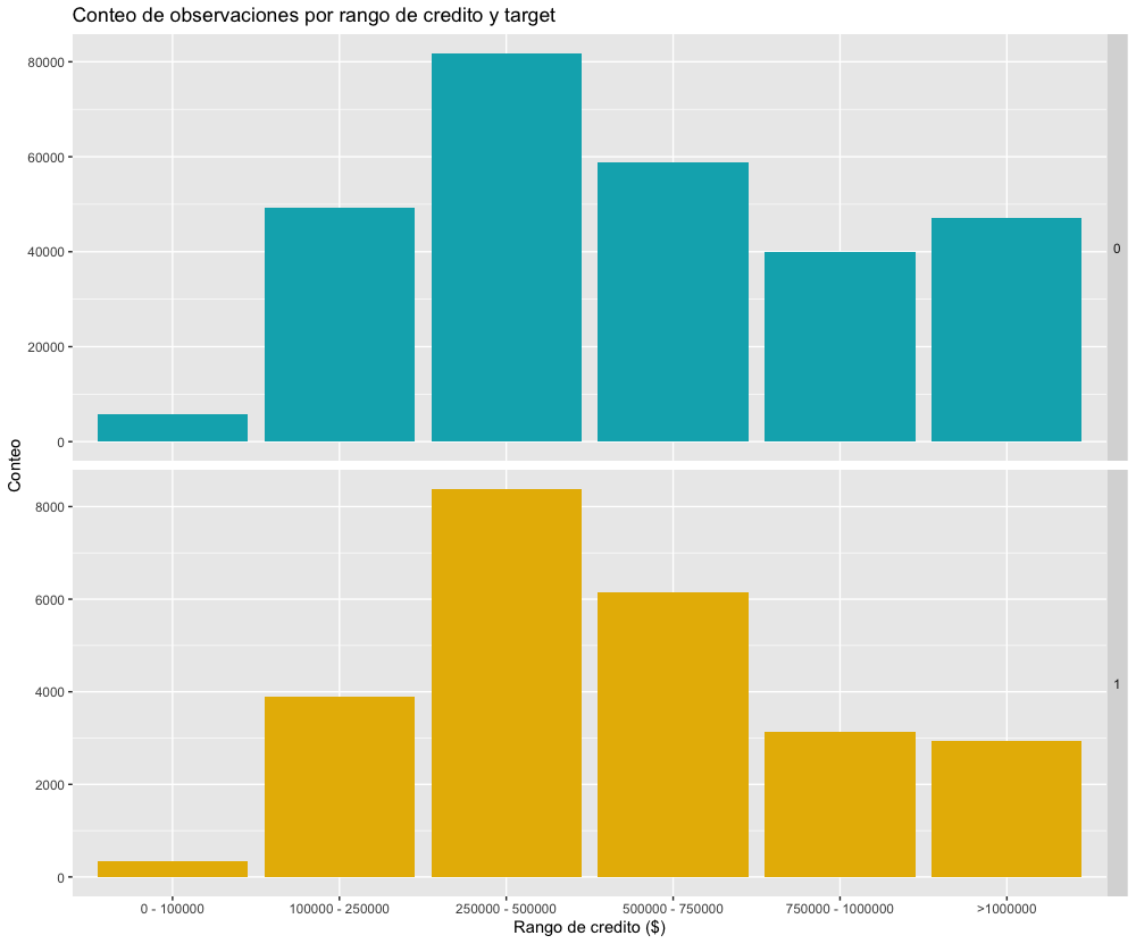
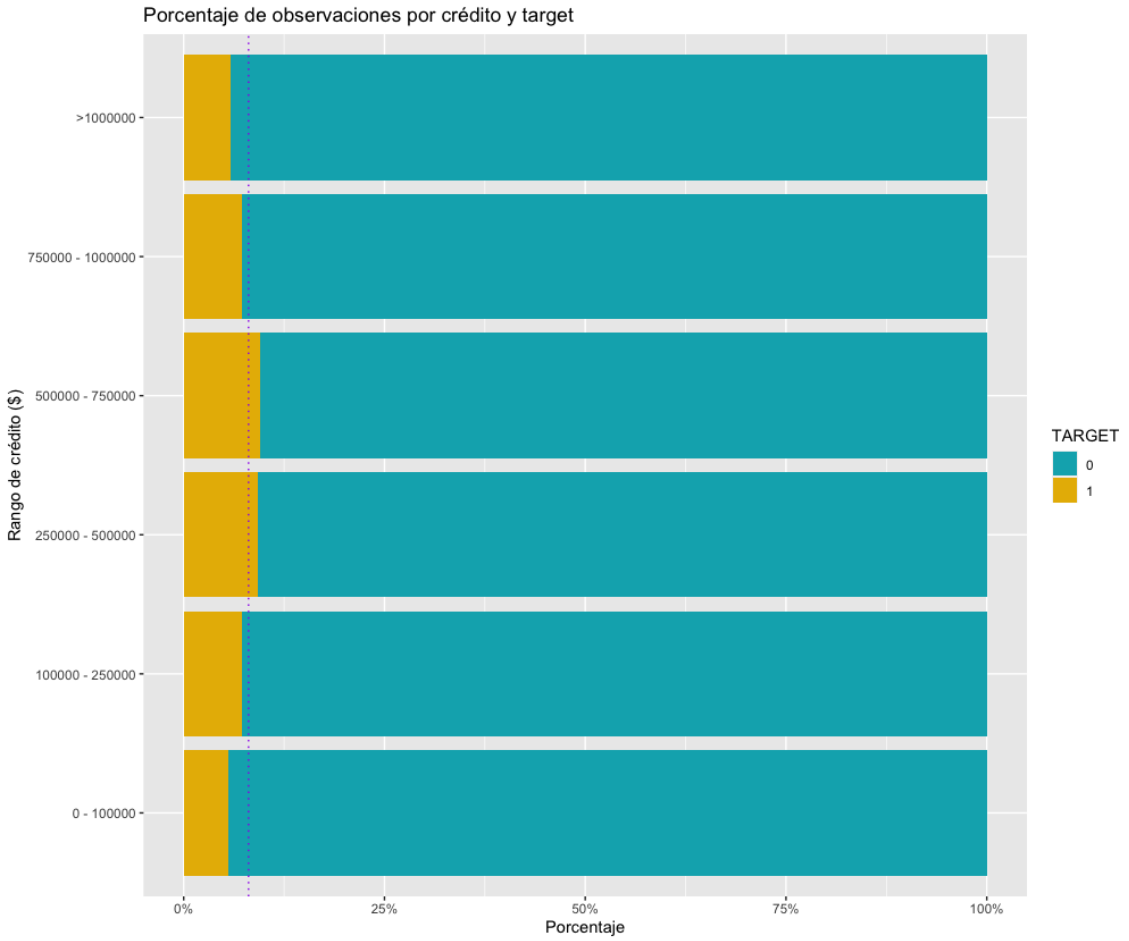


Figura 11: Porcentaje de observaciones por crédito y clase, HCDR



La figura 11 muestra el porcentaje de observaciones de cada clase para cada rango de crédito solicitado (*AMT_CREDIT*). En este caso se observa que en el rango $250.000 a $750.000 la cantidad de solicitantes de la clase minoritaria es mayor al promedio. En el rango 0 a $100.000 la cantidad de solicitantes de la clase minoritaria es menor al promedio mientras que lo mismo sucede para valores mayores a $1.000.000 .

Figura 12: Conteo de observaciones por tipo de ingreso y clase, HCDR

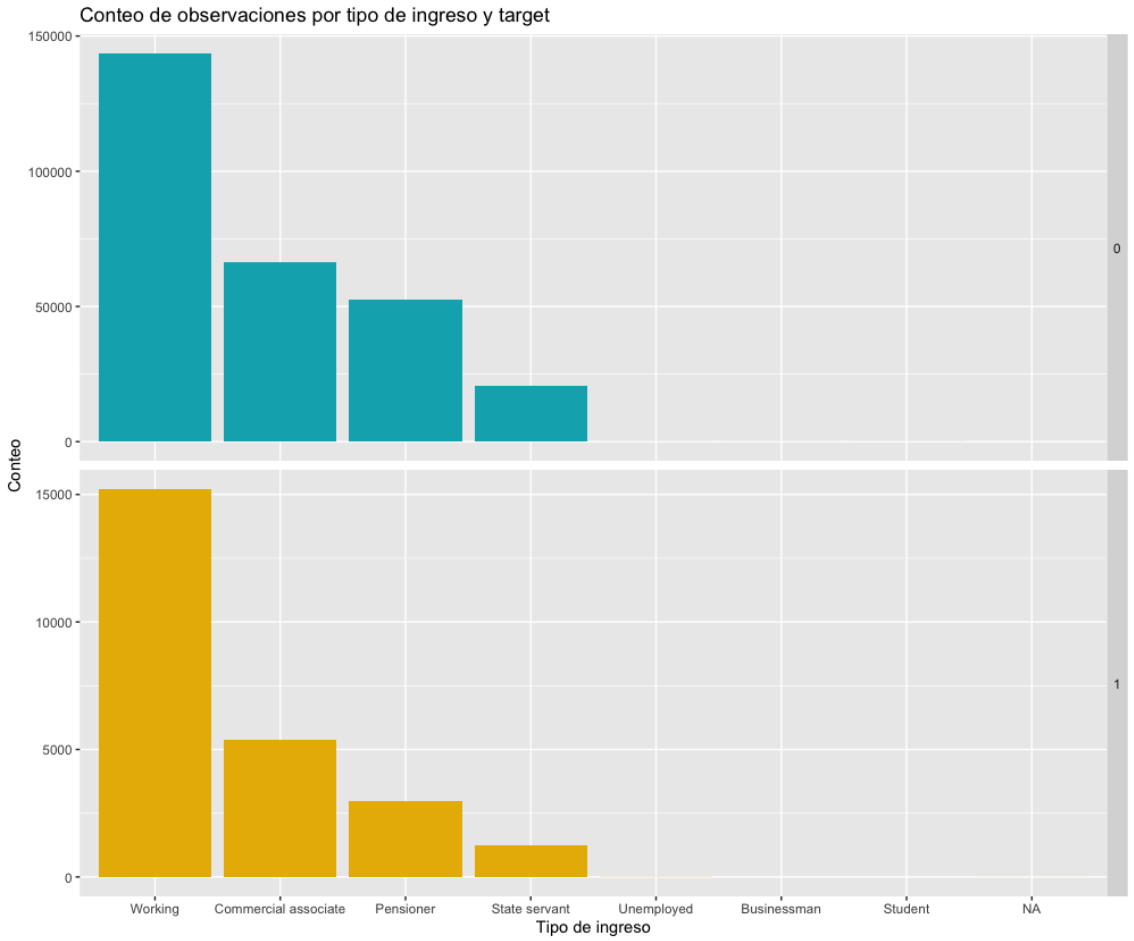
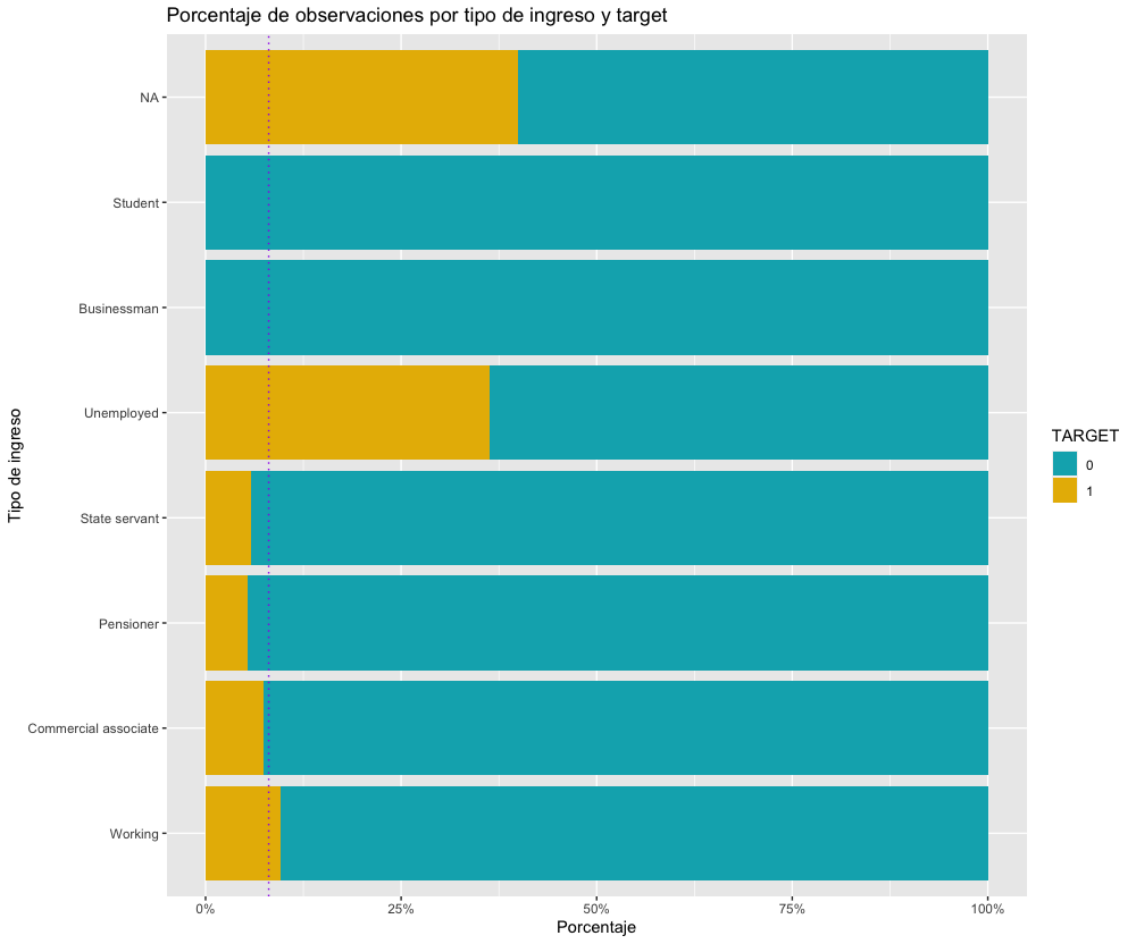


Figura 13: Porcentaje de observaciones por tipo de ingreso y clase, HCDR



La figura 12 exhibe el conteo de observaciones por tipo de ingreso según la clase *TARGET*. Se observa que la gran mayoría de solicitantes pertenecen a 4 categorías (“Working”, "Commercial associate", "Pensioner", "State servant"). El resto de las categorías no muestran una presencia significativa en los datos. Sin embargo, en la figura 13 se observa que la categoría “Unemployed” muestra una gran proporción de la clase minoritaria por encima del promedio y, por el contrario, las clases “Businessman” y “Student” muestran una proporción muy alta de la clase mayoritaria. La clase de mayor presencia (“Working”) presenta una proporción de clase minoritaria mayor al promedio mientras que "Commercial associate", "Pensioner" y "State servant" muestran proporciones de la clase minoritaria por debajo del promedio. Es importante destacar la gran proporción de clase minoritaria que se encuentra en los valores ausentes “NA”.

En el caso de la base de datos HCDR, al contar con 122 variables un gráfico de matriz de correlación no ayudaría a entender estas relaciones. En el Anexo 2- “Tabla de correlaciones absolutas mayores a 0,7, HCDR” se encuentra una tabla con los 126 pares de variables que presentan correlación absoluta mayor a 0,7. Estos 126 pares están conformados por 81 variables únicas por lo que se puede concluir que existe una gran proporción de las variables con alta correlación. En este sentido, la selección de las variables se vuelve importante a la hora de entrenar los modelos.

De las variables más importantes se destacan *AMT_CREDIT* y *AMT_GOODS_PRICE* con correlación 0,99. En contexto, esto podría explicarse porque ambas variables indican que el crédito otorgado y el bien declarado que se adquiere tras recibir el crédito es, en su mayoría, del mismo monto. Otro par relevante es *AMT_CREDIT* y *AMT_ANNUITY* con correlación 0,77.

METODOLOGÍA

Ante la necesidad de un marco comparativo que exhiba la implicancia que tienen distintas técnicas de machine learning sobre la reducción del error predictivo, a continuación se detallan todas las variaciones metodológicas llevadas a cabo en el proceso de experimentación.

El proceso que se llevará a cabo en este trabajo consiste en desarrollar múltiples modelos simples con distintos tipos de métodos de selección y tratamiento de variables sobre 3 métricas estadísticas con distinta influencia de negocios. A su vez, a estos modelos se les incorporarán distintos métodos de balanceo de clase. Los métodos que mejor reflejen una mayor performance teniendo en cuenta las 3 métricas serán incorporados a modelos más complejos. Luego, manteniendo los mejores tratamientos, se aplicarán modelos que busquen minimizar los costos asociados teniendo en cuenta costos fijos y variables por observación. Finalmente se aplicarán los mismos tratamientos sobre modelos que buscarán optimizar los costos variables al reemplazar la matriz de costos y alterar el punto de corte en la toma de decisiones observando también el impacto en las 3 métricas originales y sus implicancias.

TRATAMIENTOS Y SELECCIÓN DE VARIABLES

Para ambas bases de datos se realizarán 5 versiones de tratamiento y selección de variables teniendo en cuenta algunas características mencionadas a partir de las particularidades de los datos:

Tabla 2: Tratamientos GMSC

GMSC	NA(s)	Valores aislados (outliers)	Filtro de vars	Escalar	Transformaciones
V1	Casos completos	No	No	No	No
V2	Valores promedios	No	No	No	No
V3	Valor promedio (NumberOfDependents)	No	No	No	No
V4	(mejor V1-V3)	Si	No	Si	Si
V5	(mejor V1-V3)	Si	Si	Si	Si

En el caso de GMSC, la versión 1 utilizará los datos sin tratamiento alguno, es decir, tomará los datos tal y como vienen dados de origen. El único tratamiento que se realizará es sobre los valores ausentes, ya que ante la presencia de valores ausentes los modelos presentan errores. En este sentido, esta primera versión eliminará los casos incompletos (las filas que cuenten con algún valor ausente). Las versiones 2 y 3 presentarán las mismas condiciones alterando el tratamiento de valores ausentes. La versión 4 introducirá nuevos tratamientos y finalmente la versión 5 intentará replicar la versión anterior con una reducción de dimensionalidad (eliminar variables para reducir el ruido).

NA(s) / Valores ausentes: Como ya fue mencionado, los valores ausentes son un problema persistente en el análisis de datos debido a que los modelos no pueden procesarlos sin antes ser tratados. En este sentido se utilizarán 3 estrategias:

1. Casos completos: Elimina las observaciones que presentan valores ausentes.

   Resultado: 120.269 observaciones (150.000 observaciones originales)

2. Valores promedios: Reemplaza los valores ausentes por el valor promedio de la columna.

   Resultado: No elimina observaciones

3. Valores promedios (*NumberofDependents)*: Reemplaza los valores ausentes por el valor promedio de la columna en la variable *NumberofDependents* (menor cantidad de valores ausentes) y posteriormente elimina las observaciones con valores ausentes en la variable *MonthlyIncome* (mayor cantidad de valores ausentes.

   Resultado: 120.269 observaciones (150.000 observaciones originales)

Dado que las variables que contienen datos ausentes son variables numéricas es posible reemplazar dichos valores con valores promedios. En el caso de la variable que indica la cantidad de familiares dependientes, se toma el valor promedio a pesar de que, conceptualmente, esta variable toma valores enteros. Las versiones 4 y 5 tomarán la estrategia de valores ausentes que mejor optimicen las métricas objetivo entre las versiones 1, 2 y 3.

Valores aislados (outliers): Los valores aislados son observaciones extremas dentro de una serie de datos que pueden afectar la estimación de los modelos. Estos casos deben ser tratados para intentar reducir el ruido en los modelos. Es importante destacar que existen varios métodos para considerar los casos aislados y su tratamiento posterior. En este trabajo se considerarán casos aislados para este set de datos a los datos más extremos en la distribución de dos variables en particular:

- *RevolvingUtilizationOfUnsecuredLines:* Se eliminará el caso más extremo con un valor de 50.708 .
- *DebtRatio*: Se eliminarán los casos superiores a 200.000 (4/150.000). Se toma esta decisión ya que si se hubiese implementado la regla de “q3 + 1,5 x IQR”, esto hubiese eliminado 20% de los casos en la variable *DebtRatio*.

Filtro de variables: El uso de las 10 variables independientes (atributos) del set de datos será implementado en las primeras 4 versiones mientras que en la última versión se realizará una selección reducida que cuente con las variables de mayor importancia en los modelos realizados anteriormente. Con la reducción de variables en la versión 5 lo que se buscará es probar si con una menor cantidad de atributos es posible reducir el error estadístico de los modelos. Para eso es necesario determinar cuáles son las variables de mayor importancia.

El método para determinar las variables de mayor importancia que se utilizará tiene el siguiente procedimiento: tomando la versión que obtenga la métrica AUC mayor entre las primeras 4 versiones se calculará la métrica *MeanDecreaseAccuracy* (“Disminución de precisión promedio”) sobre los modelos Random Forest (que fueron los mejores modelos) con distintos métodos de balanceo de clases. Esta métrica indica la incidencia de cada variable independiente incorporada en el modelo sobre la precisión del mismo. Se calculará la métrica sobre los modelos Random Forest asumiendo que estos modelos obtendrán mejores resultados de AUC superando a los modelos de regresión logística.

Para cada modelo Random Forest se tomarán las 5 variables que contengan mayor valor *MeanDecreaseAccuracy* y se las ordenará de mayor a menor valor,

dando un valor de 5 a la variable de mayor importancia y de 1 a la de menor importancia. Las 5 variables cuyo valor agregado (tomando todos los modelos Random Forest) sea mayor serán tenidas en cuenta en la versión 5 mientras que las 5 variables restantes serán omitidas. La siguiente tabla muestra el procedimiento:

Tabla 3: Selección de variables GMSC

GSMC	Balanceo #1	Balanceo #2	Balanceo #3	Balanceo #4	Balanceo #5
MeanDecreaseAccuracy #1 (Valor 5)	Variable A	Variable A	Variable A	Variable A	Variable A
MeanDecreaseAccuracy #2 (Valor 4)	Variable B	Variable B	Variable C	Variable B	Variable G
MeanDecreaseAccuracy #3 (Valor 3)	Variable C	Variable C	Variable B	Variable C	Variable B
MeanDecreaseAccuracy #4 (Valor 2)	Variable D	Variable F	Variable D	Variable D	Variable D
MeanDecreaseAccuracy #5 (Valor 1)	Variable E	Variable G	Variable G	Variable E	Variable C

En el ejemplo anterior la variable A tomaría una puntuación total de 25 (5+5+5+5+5), mientras que la variable G tomaría una puntuación total de 6 (0+1+1+0+4). Las variables que serían tomadas en cuenta para la versión 5 a partir de estos resultados serían: variable A (25), variable B (18), variable C (14), variable D (8) y la variable G (6). Las variables E y F serían descartadas al igual que el resto de las variables que no forman parte del cuadro.

Esta metodología busca quedarse con las mejores variables promediando la influencia sobre varios modelos que se diferencian en el balanceo de clases ya que cada modelo puede presentar una influencia particular por parte de sus variables independientes. La métrica *MeanDecreaseAccuracy* exhibe la medida en que una variable mejora la precisión del bosque al predecir la clasificación en el modelo de Random Forest. [1]

Escalar: Al utilizar algoritmos de regresión puede ser necesario rescalar variables numéricas para mejorar la precisión de los modelos y reducir el tiempo de procesamiento [2]. Este procedimiento tiene sentido al aplicarse sobre variables numéricas con valores de alto rango. Las versiones 1,2 y 3 no serán alcanzadas por esta metodología, mientras que las versiones 4 y 5 serán tratadas a través de la estandarización de algunas variables como

[1] https://wiki.q-researchsoftware.com/wiki/Machine_Learning_-_Random_Forest

[2] https://analyticsindiamag.com/why-data-scaling-is-important-in-machine-learning-how-to-effectively-do-it/#:~:text=Scaling%20the%20target%20value%20is,learn%20and%20understand%20the%20problem.&text=Scaling%20of%20the%20data%20comes,algorithms%20in%20the%20data%20set.

RevolvingUtilizationOfUnsecuredLines, *DebtRatio* y *MonthlyIncome*. La fórmula de estandarización aplicada será la siguiente:

$$Z = (X - X_{prom}) / \sigma$$

Donde Z representa el valor estandarizado y σ representa el desvío estándar de la muestra.

Transformaciones: Se incorporarán 3 variables a las versiones 4 y 5 :

- *SumTimesPastDue*: *NumberOfTime30.59DaysPastDueNotWorse* + *NumberOfTimes90DaysLate* + *NumberOfTime60.89DaysPastDueNotWorse*
- *SumTimesPastDueOneOrMore*: Toma valor 1 si *SumTimesPastDue* es mayor o igual a 1, caso contrario toma valor 0.
- *HighRevolvingUtilization*: Toma valor 1 si *RevolvingUtilizationOfUnsecuredLines* es mayor o igual a 0,645, caso contrario toma valor 0.

Tabla 4: Tratamientos HCDR

HCDR	NA(s)	One Hot Encoding	Variables combinadas	Filtro de Vars.	Escalar	Transformaciones
V1	<50.000+ casos completos	No	Si	No	No	No
V2	<50.000+ casos completos	Si	Si	No	No	No
V3	<50.000+ casos completos	No	Si	Si	No	No
V4	Valores promedios	Si	No	Si	Si	No
V5	<150.000+ valores promedios	Si	No	Si	Si	Si

Para la serie de datos HCDR, la versión 1 tomará los datos de origen realizando tratamiento sobre los valores ausentes y creando variables combinatorias. La versión 2 aplicará el método *One Hot Encoding*. La versión 3 realizará un filtro de variables para reducir en gran medida la dimensionalidad de los datos. La versión 4 omitirá el uso de variables combinatorias, realizará un re-escalamiento de algunas variables y reemplazará los valores ausentes por valores promedio. Por último la versión 5 introducirá transformaciones en las variables y un método híbrido para tratar los valores ausentes.

NA(s) / Valores ausentes: Dada la presencia de una alta cantidad de valores ausentes en cantidad de observaciones como también en términos de variables, se utilizarán 3 estrategias:

1. <50.000+ casos completos: Elimina las variables con más de 50.000 valores ausentes, luego filtra los casos incompletos.

   Resultado: -38% de las variables, -14% de las observaciones

2. Valores promedios: Reemplaza los valores ausentes por los valores promedios de la variable.

   Resultado: Sin pérdida de información.

3. <150.000 + valores promedios: Elimina las variables con más de 150.000 valores ausentes, luego reemplaza los valores ausentes por los valores promedios.

4. Resultado: -16% de las variables.

One Hot Encoding: Dada la gran cantidad de variables no numéricas en el set de datos es necesario implementar una estrategia que permita procesar estos datos por parte de los modelos que se implementarán. La estrategia *One Hot Encoding* crea nuevas variables para cada valor distinto que se encuentra en la variable aplicada y se imputa un “1” en la columna de ese registro, un “0” en las que no. Como ejemplo, en el caso hipotético de una columna que indique el género, si la columna posee como valores “hombre” y “mujer”, la estrategia crea dos nuevas variables en las que indica con “1” en la columna de “hombre” si la observación indicaba “hombre” en la columna original, y un “0” en la columna de “mujer”.

La versión 1 no ejecutará la estrategia de *One Hot Encoding* (esto podría tener complicaciones en la ejecución de los modelos como así también en los métodos de balanceo de clase). La versión 2 lo implementará sobre las variables no numéricas luego de la separación entre datos de entrenamiento y testeo. La versión 3 tampoco ejecutará el método dado que el resto de los tratamientos producirán variables numéricas. Las versiones 4 y 5 ejecutarán el método antes de la separación entre datos de entrenamiento y testeo sobre las variables no numéricas.

Es importante tener en cuenta que al realizar el método *One Hot Encoding* luego de separar los datos es posible que la cantidad de variables en ambos set de datos quede desigual ya que el método crea nuevas variables en

relación a la cantidad de factores que contengan. En este sentido, será necesario crear las variables faltantes para equiparar este desbalance.

Variables combinadas: Este método tiene como objetivo reducir la dimensionalidad en un set de datos con una gran cantidad de variables categóricas. Este tratamiento combina la información de un conjunto de variables categóricas en un valor numérico a través de una regresión lineal. Al obtener un valor numérico que resulte de la combinación de las variables incluidas en la regresión (variables independientes) se puede eliminar posteriormente el conjunto de variables y así reducir la dimensionalidad de los datos. A continuación se brinda un ejemplo del método:

1. Formula: TARGET = $Regresor_1$ + $Regresor_2$ + $Regresor_3$ [Sobre los datos de entrenamiento]
2. Regresión lineal sobre la fórmula en el paso 1
3. Predicción de valores sobre los datos de testeo
4. Eliminación de regresores 1, 2 y 3 e introducción de la nueva variable combinada.

En el ejemplo anterior se produce una reducción de 2 variables (se eliminan los tres regresores y se introduce uno nuevo). Las variables serán combinadas en relación al tipo de información que aportan. Se introducirán 4 nuevas variables combinadas:

- situation_cat: Variable que combina la información situacional en la que se genera la solicitud de préstamo (ej. *WEEKDAY_APPR_PROCESS_START* que simboliza el día de la semana en que el hecho se produce)
- location_cat: Variable que combina información acerca de la locación donde vive el solicitante.
- contact_cat: Variable que combina información sobre variables que indican si las distintas direcciones del solicitantes coinciden entre sí.
- document_cat: Variable que combina información sobre variables que indican si el solicitante proporcionó documentos.

Con este proceso se introducirán 4 nuevas variables combinadas y se eliminarán 40 variables.

Filtro de variables: Las versiones 1 y 2 solamente omitirán las variables que formen parte del tratamiento de valores ausentes y variables combinatorias. La versión 3 seleccionará las mejores 10 variables sobre el mejor modelo resultante de las versiones 1 y 2 utilizando la misma metodología del MeanDecreaseAccuracy mencionada anteriormente. En las versiones 4 y 5 se

eliminarán variables que presenten correlación mayor a 0,7 (valor absoluto) con respecto a otras variables.

Escalar: Al igual que en el set de datos GMSC, ciertas versiones de este set de datos serán alcanzadas por el método de estandarización de variables numéricas. Las versiones 1, 2 y 3 no contarán con esta metodología mientras que las versiones 4 y 5 serán tratadas por el método de estandarización sobre las variables numéricas que no fueron previamente impactadas por el método One Hot Encoding.

Transformaciones: Se incorporarán 6 nuevas variables a la versión 5 que intentarán captar nueva información sobre la capacidad de repago del crédito solicitado:

- *credit_over_income = AMT_CREDIT / AMT_INCOME_TOTAL*
- *annuity_over_income = AMT_ANNUITY / AMT_INCOME_TOTAL*
- *goods_over_income = AMT_GOODS_PRICE / AMT_INCOME_TOTAL*
- *annuity_over_goods = AMT_ANNUITY / AMT_GOODS_PRICE*
- *goods_over_credit = AMT_GOODS_PRICE / AMT_CREDIT*
- *credit_minus_goods = AMT_CREDIT - AMT_GOODS_PRICE*

BALANCEO DE CLASES

Dado el gran desbalance de clases presente en las variables objetivo de ambos set de datos se introducirán métodos de rebalanceo para que los modelos predictivos cuenten con mayor información sobre los casos a predecir (solicitantes con problemas de repago crediticio). En este sentido, para cada tratamiento de variables se elaborarán modelos con 5 métodos de rebalanceo distintos, siendo uno de ellos el caso en que no se trata esta problemática y se utilicen los datos desbalanceados como los presenta el set de datos originalmente para poder exhibir los resultados en la aplicación de esta metodología. Los otros 4 métodos son los siguientes:

- Over Sampling: Introducción de sesgo duplicando muestras aleatorias de la clase minoritaria.
- Under Sampling: Introducción de sesgo eliminando muestras aleatorias de la clase mayoritaria.
- Over-Under Sampling: Técnica mixta que combina ambas técnicas anteriores.
- Smote: Técnica de sesgo duplicando muestras de la clase minoritaria en forma sintética. Esta técnica consiste en detectar observaciones que

pertenecen a la clase minoritaria cuyos datos se asemejan para luego crear observaciones sintéticas con valores intermedios.[3]

En el caso de Over Sampling el resultado será aumentar la cantidad de observaciones igualando la distribución de clases. Para Under Sampling el resultado será disminuir la cantidad de observaciones igualando la distribución de clases. Para Over Under Sampling se igualará la distribución de clases sin modificar la cantidad de observaciones. Para la técnica Smote se reducirán la cantidad de observaciones y la distribución de clases se equiparan sin ser estas exactamente iguales.

Múltiples modelos predictivos serán evaluados para ambos set de datos a lo largo de este trabajo. Los métodos de rebalanceo de clase mencionados serán evaluados y los mejores serán utilizados para introducir modelos de optimización de costos.

MÉTRICAS

Las 4 métricas que se utilizarán a lo largo de este trabajo serán:

- Accuracy (precisión global): Exhibe la cantidad de predicciones correctas sobre la totalidad de predicciones realizadas. Esta métrica, probablemente la más común a la hora de evaluar modelos de clasificación[4], permite reconocer de forma clara el error predictivo de los modelos (Error = 1 – accuracy). Sin embargo, al tratar problemas de clasificación sobre datos desbalanceados esta métrica puede ser poco útil ya que no indica la capacidad de los modelos en predecir correctamente los distintos valores de la variable dependiente. Dicho esto, comprender el intercambio entre accuracy y otras métricas alterando la métrica objetivo de los modelos predictivos es de gran valor para el análisis de otorgamiento de créditos.
- AUC (Área bajo la curva de ROC): Esta métrica de performance, a diferencia de accuracy, exhibe que tan buena es la predicción en un modelo de clasificación teniendo en cuenta la capacidad de predecir correctamente los casos positivos como así también los casos negativos. Por otro lado, dicha métrica pondera de la misma manera ambos tipos de errores y es insensible al punto de corte, siendo la misma poco ilustrativa en problemas como el crediticio donde los tipos de errores presentan costos desiguales. A su vez, AUC no es

[3] https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
[4] https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/

considerada una buena métrica para casos en los que la clase minoritaria es la clase positiva .[5]

- F1-Score: Esta métrica es una media armonizada de las medidas de precisión en positivos y sensibilidad (“precision” y “recall”). La métrica “precision” (precisión en positivos) es una medida que indica qué porcentaje de las predicciones positivas del modelo son correctas. Se calcula como la relación entre el número de verdaderos positivos y el número total de instancias positivas predichas por el modelo. La métrica “recall” es una medida que indica qué porcentaje de las instancias positivas reales fueron detectadas correctamente por el modelo. Se calcula como la relación entre el número de verdaderos positivos y el número total de instancias positivas reales en el conjunto de datos. Es importante mencionar que “precision” y “recall” son dos medidas complementarias, un modelo puede tener una alta precisión pero baja recall o viceversa. Es por esto que F1-Score es una métrica relevante dadas las características desbalanceadas de los datos del problema.
- Costo Promedio: Costo promedio al que las instituciones financieras tienen que incurrir considerando los resultados del modelo predictivo. Se elige implementar el promedio por solicitante a diferencia del costo total dado que cada modelo podría trabajar sobre una cantidad distinta de solicitudes.

Dado que uno de los objetivos del trabajo consiste en comparar el rendimiento de varios modelos y métricas, la inclusión de accuracy, AUC y F1-Score como se explica por ser medidas comúnmente utilizadas en la literatura para evaluar modelos de clasificación binaria. Existen métricas como PR AUC (métrica que muestra la relación entre precisión y recall) que, a priori, pueden tener mayor sentido en relación al problema de desbalance de clases. Sin embargo, teniendo en cuenta que ambas competencias Kaggle de las cuales fueron extraídos los datos del trabajo tienen como objetivo puntual maximizar la métrica AUC, se incluyó esta última a modo comparativo.

Por otro lado, la métrica de negocios más relevante es la de costo promedio, que ayudará a definir cuales son los modelos y técnicas que tienen mejores resultados e ilustrar el trade-off entre dicha métrica y accuracy.

[5] B.K. Baloch et. al (2019)

Modelos

MODELOS INSENSIBLES AL COSTO

Para cada versión descrita anteriormente se implementarán una serie de modelos insensibles al costo. La estructura de los modelos empleados será la siguiente para versión en ambos set de datos:

1. Regresión Logística optimizando la métrica accuracy.
2. Regresión Logística con validación cruzada optimizando la métrica F1-Score con y sin métodos de rebalanceo.
3. Bosques Aleatorios (“Random Forests”) con validación cruzada optimizando la métrica AUC con y sin métodos de rebalanceo.
4. XGBoost optimizando la métrica AUC con y sin métodos de rebalanceo.
5. XGBoost optimizando la métrica AUC con validación cruzada, búsqueda de hiper parámetros con y sin métodos de rebalanceo.

Esta estructura conlleva comparar cientos de modelos entrenados teniendo en cuenta los distintos algoritmos, técnicas de rebalanceo y versiones de datos. En este sentido se buscará exhibir el impacto en las distintas métricas objetivo utilizando algoritmos de distintas categorías. La regresión logística es un algoritmo regresivo simple que es muy utilizado en el contexto de análisis de datos. En cambio los modelos Random Forest y XGBoost son algoritmos de ensamble con mayor complejidad. La validación cruzada permite iterar los modelos en diferentes particiones de los datos intentando generar resultados que sean independientes a los datos utilizados en el entrenamiento. Finalmente, los modelos XGBoost serán evaluados con iteraciones sobre sus hiper parámetros para intentar mejorar la métrica objetivo.

MODELOS COST-SENSITIVE

En el campo del credit scoring, los modelos tradicionales de machine learning suelen tener dificultades para capturar correctamente los casos en los que los solicitantes presentan dificultades de repago debido a la presencia de clases desbalanceadas en los datos. Esto significa que una de las clases tiene una cantidad significativamente mayor de instancias que la otra. La razón principal por la cual los modelos tradicionales tienen dificultades con estos casos es que suelen optimizar las métricas tradicionales como accuracy, F1-Score, y AUC.

Sin embargo, estas métricas tradicionales no son suficientes para llevar los modelos a la práctica en los casos de clases desbalanceadas. Esto se debe

a que estas métricas no tienen en cuenta los costos asociados con las diferentes clases. Por ejemplo, en un caso de clases desbalanceadas donde la clase negativa es significativamente mayor que la clase positiva, un modelo que maximiza la accuracy o el F1-Score puede no ser la mejor opción en términos de costos. El modelo puede tener una alta “precision”, pero un bajo “recall”, lo que resulta en un gran número de casos positivos no detectados.

Por lo tanto, es necesario considerar los costos asociados con las diferentes clases a la hora de elegir un modelo de credit scoring. Una forma de hacerlo es utilizar métricas que tengan en cuenta los costos, como el costo promedio. Estas métricas permiten a los modelos tener un mejor equilibrio entre “precision” y “recall”, lo que resulta en una mejor eficiencia en términos de costos.

Los modelos y métricas detallados anteriormente en la presente sección son de gran utilidad en la práctica de análisis de datos pero ninguno de ellos involucra a los costos. Las instituciones financieras como agentes económicos tienen la necesidad de generar rentabilidad a través de la maximización de ingresos y/o la minimización de los costos asociados al negocio. Si se tomaran los ingresos como costos negativos el desafío de maximizar la rentabilidad podría resumirse en la minimización de los costos.

Para poder visualizar la importancia de los costos por sobre el resto de las métricas en este contexto de negocios es importante ilustrar el siguiente ejemplo: Tomando los datos de la serie GMSC, un modelo sencillo de clasificación podría obtener, en promedio, una tasa de acierto en detectar buenos pagadores de malos pagadores del 93,3%. Este resultado que podría ser engañoso proviene de un modelo que acepta todas las solicitudes de crédito que se le presentan, por lo que la tasa de error del 6,7% en promedio se genera por la cantidad de malos pagadores en la muestra. Si esto fuera un modelo atractivo económicamente para una institución crediticia significa que cualquier solicitud de crédito sería aceptada sin ninguna fricción por el mercado de créditos. Este escenario no se asemeja a la realidad ya que los costos resultantes de la toma de decisiones son distintos.

Existen 4 escenarios posibles cuando una institución financiera se enfrenta a una solicitud crediticia:

Tabla 5: Escenarios generales en credit scoring

	Realidad (Buen pagador)	*Realidad (Mal pagador)*
Predecir (Buen pagador)	Crédito otorgado y lo repaga (VN)	Crédito otorgado y no se repaga (FN)
Predecir (Mal pagador)	Crédito rechazado que podría haber sido repagado (FP)	Crédito rechazado que no habría sido repagado (VP)

Lo que se puede percibir de estos 4 escenarios que muestra el cuadro anterior es que el caso ideal es donde el crédito se otorga y el solicitante está en condiciones de repagarlo para el cual la institución financiera estaría generando ganancias a través de ese préstamo. El caso siguiente es el escenario en que no se otorga el crédito a un mal pagador teniendo un impacto económico nulo. Los casos en los que la predicción no es correcta representan un costo para la institución aunque estos no son simétricos. Es esperable que el peor de los casos sea otorgar el crédito a un mal pagador mientras que no otorgar un crédito a un buen pagador se podría considerar un costo de oportunidad, es decir, un costo al fin pero no mayor al escenario opuesto.

Costos fijos

Teniendo en cuenta estos escenarios, una primera aproximación al problema podría asumir costos fijos que muestran la siguiente asimetría:

Tabla 6: Matriz de confusión con costos fijos

	Realidad (Buen pagador)	*Realidad (Mal pagador)*
Predecir (Buen pagador)	-1	5
Predecir (Mal pagador)	1	0

Como se puede notar en el cuadro anterior, el escenario en donde se otorga el crédito al solicitante que está en condiciones de repagarlo se denota con un costo negativo, haciendo referencia a que la institución crediticia genera beneficios en esos casos. Desafortunadamente, la librería MLR de R Studio no permite imputar costos negativos en sus modelos por lo que es conveniente asumir que estos costos sean nulos en lugar de ser negativos.

Tabla 7: Matriz de confusión con costos fijos, con valor VN nulo (0).

	Realidad (Buen pagador)	*Realidad (Mal pagador)*
Predecir (Buen pagador)	0	5
Predecir (Mal pagador)	1	0

Una vez establecidos los costos de los escenarios que surgen de la toma de decisión (aceptar o rechazar una solicitud de crédito) es necesario introducir la noción de punto de corte (o "*threshold*"). El punto de corte hace referencia al nivel de probabilidad necesario para tomar la decisión de aceptar o rechazar la solicitud. Los modelos en su gran mayoría traen consigo por default el punto de corte asignado en 0,5. Es decir, si el modelo indica que un solicitante se encuentra en condiciones de repagar su crédito con probabilidad 0,6, la decisión final será otorgar el crédito ya que es superior al punto de corte. Si la probabilidad de repago fuera de 0,4 entonces la decisión será de rechazar la solicitud.

El punto de corte es un parámetro más dentro del modelo de decisión ya que modificando el mismo los resultados económicos cambian sustancialmente. El cuadro anterior muestra que el punto de corte debería tomar en cuenta la asimetría en costos para poder optimizar la decisión. Una primer aproximación a este concepto es poder imputar el punto de corte teórico al proceso de decisión:

$$\text{Theoretical Threshold} = FN / (FN + FP) = 0{,}83$$

Este punto de corte teórico indica que para aceptar un crédito la probabilidad del modelo debería ser superior a 0,83 ya que el costo de aceptar un crédito malo es superior al costo de no aceptar un crédito bueno (asumiendo que los costos VN y VP son nulos). Por lo que podríamos esperar que este punto de corte generará menores costos que el punto de corte por default de 0,5 .

Costos por observación ("Example dependent")

Otra manera de aproximarse al problema es asumir que los costos para cada solicitante son distintos. Este supuesto se asemeja más al caso real ya que el tamaño del crédito varía en relación a cada solicitante por lo que los costos para cada uno de ellos no serían fijos. Dado que las bases de datos utilizadas en este trabajo no proveen información acerca de los costos en

los que incurren las instituciones financieras relacionadas es necesario deducir los mismos a través de supuestos y la información disponible.

Para poder estimar los costos se tomarán tres supuestos en base a los parámetros más importantes. En primer lugar se asume que la tasa de interés que pagan los solicitantes por sus créditos es la misma para todos los solicitantes y en ambas muestras (4% anual). En segundo lugar se asume que el plazo temporal del crédito es de un año. Por último, para el caso de la muestra GMSC en la que el valor solicitado por cada cliente no es proporcionado se asume que el crédito tomado representa la mitad del ingreso mensual de cada observación (expresado en la variable *MonthlyIncome*), mientras que para la muestra HCDR este valor viene expresado en la variable *AMT_CREDIT*. En base a estos supuestos se obtienen los siguientes parámetros:

- r = 0,04 (tasa de interés anual para todos los créditos)
- (GMSC) loan = 0,5 x *MonthlyIncome*
- (HCDR) loan = *AMT_CREDIT*

Es importante destacar que al modificar la matriz de costos utilizando costos por observación en lugar de costos fijos la escala de costos resultantes cambie significativamente. Dado que el objetivo no es comparar los costos de un modelo basado en costos fijos versus un modelo basado en costos por observación, sino más bien comparar los costos entre modelos aplicados a la misma serie de datos con distintos marcos para estimar los costos pero de forma independiente, este resultante no sería un inconveniente. Con estos parámetros podemos estimar la fórmula de los costos asociados a los 4 escenarios posibles:

Tabla 8: Matriz de confusión example dependent

	Realidad (Buen pagador)	*Realidad (Mal pagador)*
Predecir (Buen pagador)	(-) loan x r	loan x (1+r)
Predecir (Mal pagador)	loan x r	0

La tabla 8 muestra que el costo de los casos falsos positivos (FP) es básicamente el costo de oportunidad del capital prestado a lo largo de un año. El costo en el caso de los falsos negativos (FN) asume que el capital prestado no se recupera en su totalidad y que además ese capital podría haber sido prestado a otro solicitante (costo de oportunidad). El costo en el caso de los verdaderos positivos (VP) es nulo y el caso de los verdaderos negativos (VN) es negativo siendo este el interés generado en un buen préstamo.

Al igual que en los modelos de costos fijos, la librería MLR de R Studio presenta inconvenientes a la hora de imputar costos negativos en los modelos "*example dependent*". En este sentido, una posible alternativa es asumir que el costo del verdadero negativo es cero, aunque esto no considera el hecho de que el banco ganaría intereses. Si se quisiera considerar esto, la forma de hacerlo sería optimizar el punto de corte de forma empírica (tal y como se presentará en la siguiente sección "*Costos por observación + threshold empírico óptimo"*).

Tabla 9: Matriz de confusión example dependent, con valor VN nulo (0).

	Realidad (Buen pagador)	*Realidad (Mal pagador)*
Predecir (Buen pagador)	0	loan x (1+r)
Predecir (Mal pagador)	loan x r	0

Adicionalmente, a pesar de que los costos por observación parecen una mejor estimación por sobre los costos fijos, la librería MLR no permite obtener las probabilidades de las predicciones por lo que no es posible alterar el punto de corte (solamente provee la clase [0,1], similar a tomar el punto de corte en 0,5).

Costos por observación + threshold empírico óptimo

Para poder superar las restricciones que presenta la librería MLR e introducir a los modelos la noción del punto de corte óptimo con costos por observación se presenta el siguiente procedimiento:

1. Elaborar un modelo inicial que genere probabilidades con el punto de corte por default en 0,5.
2. Establecer la matriz de costos por observación con el valor de los verdaderos negativos distinto de cero (- loan x r).
3. Aplicar la matriz de costos sobre las predicciones obtenidas del modelo inicial. De esta forma se obtienen los costos para cada observación según la predicción del modelo y la verdad sobre el comportamiento del solicitante.
4. Optimizar el punto de corte minimizando los costos. Introducir una función que itere sobre todos los distintos puntos de corte entre 0 y 1 buscando el valor mínimo de costos.

Con el objetivo de poder comparar las métricas entre distintos modelos del mismo set de datos (y mismo enfoque de costos) es necesario trabajar con los

costos promedios, ya que si se trabajara con los costos totales, es decir, con la suma de los costos de todas las observaciones, sería engañoso ya que cada modelo podría estar calculando costos sobre una distinta cantidad de observaciones. Nuevamente es relevante notar que la escala de valores de los costos va a variar en relación al enfoque de la matriz en cuestión. En este último caso en el que se consideran los costos verdaderos negativos no nulos, es esperable que la escala de los costos sea distinta a la que considera los costos negativos como nulos.

Este procedimiento permite a su vez estimar el costo promedio en el caso de que la función de optimización del punto de corte busque maximizar una métrica distinta como “accuracy”, iterando sobre los distintos puntos de corte entre 0 y 1. Este valor podría ser relevante ya que comparando ambos costos con distintas optimizaciones sería posible reflejar las distintas implicancias que tienen ambos enfoques sobre los efectos del negocio.

ESTRUCTURA DE MODELOS DE COSTOS

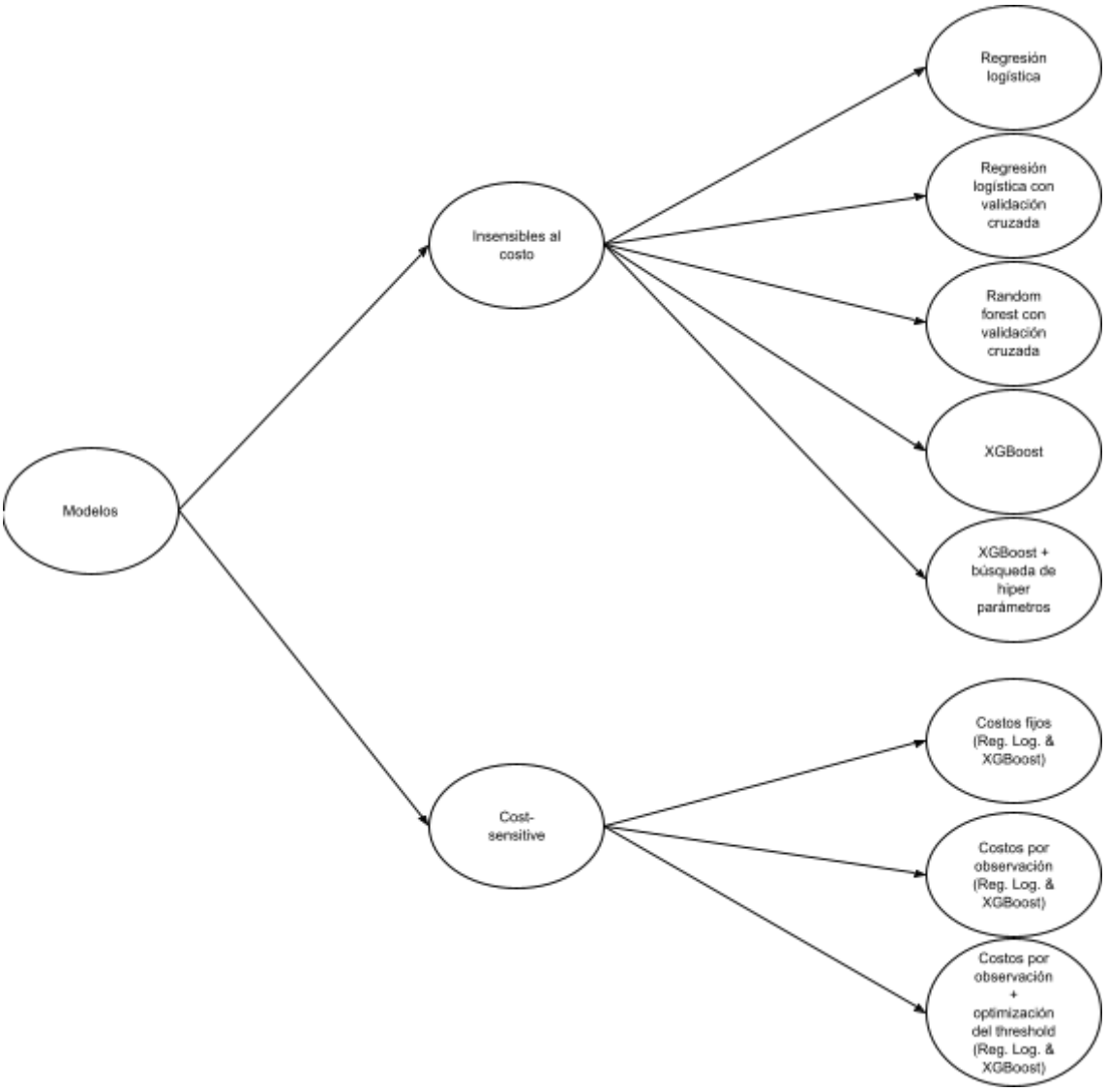
Con el objetivo de exhibir las distintas metodologías de trabajo con modelos de costos se trabajará con ambas bases de datos de la siguiente manera:

1. Costos fijos: Modelos de regresión logística, XGBoost , XGBoost optimizado. Datos con y sin rebalanceo de clases. Métricas accuracy, AUC, F1 y costo promedio.
2. Costos por observación: Modelos de regresión logística, XGBoost , XGBoost optimizado. Datos con y sin rebalanceo de clases. Métricas accuracy, AUC, F1 y costo promedio.
3. Costos por observación + threshold óptimo: Modelos de regresión logística, XGBoost , XGBoost optimizado. Datos con y sin rebalanceo de clases. Métricas Costo promedio (optimizando costos y optimizando accuracy), threshold óptimo costos y threshold óptimo accuracy.

ESTRUCTURA GENERAL DE MODELOS

La figura que se presenta a continuación resume el flujo y la estructura general de los modelos de este trabajo:

Figura 14: Estructura general de modelos



Resultados

En esta sección se presentarán los resultados de los modelos desarrollados en la sección anterior aplicados sobre ambas bases de datos. Se expondrán los resultados sobre las métricas de performance elegidas y la interpretación sobre las mismas.

MODELOS INSENSIBLES AL COSTO

Tabla 10: Regresión Logística + cross validation (GMSC)

	GMSC_V1			GMSC_V2			GMSC_V3			GMSC_V4			GMSC_V5		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
Log Reg	0,934	0,515	0,061	0,933	0,520	0,079	0,929	0,517	0,070	0,934	0,522	0,085	0,934	0,509	0,036
Log Reg +cv	0,936	0,582	0,267	0,934	0,577	0,253	0,932	0,559	0,207	0,932	0,593	0,284	0,933	0,537	0,139

La tabla 10 muestra los resultados de los primeros modelos evaluados sobre la base de datos GMSC. Lo que estos resultados representan son los valores de las métricas accuracy (“ACC”), “AUC” y “F1” para cada versión de tratamiento y selección de variables (V1-V5) implementando modelos tradicionales (“Log Reg” o Regresión Logística) y modelos ensamblados (“R. Forest” o Random Forest).

Lo primero que se puede observar de estos resultados es el hecho de que el modelo Log Reg sin rebalanceo de clases obtiene resultados muy bajos de AUC y F1 para todas las versiones. Si se observa el resultado en accuracy se puede interpretar que estos modelos están clasificando la gran mayoría de observaciones como “buenos pagadores” (*SeriousDlqin2yrs = 0*), sin importar el tratamiento, ninguno logra superar 0,522 AUC. Es interesante remarcar también el comportamiento de estos modelos cuando se agrega la validación cruzada (“Log Reg + cv”). Esta metodología produjo una mejora considerable en las métricas AUC y F1 e incluso también en accuracy (para las versiones 1, 2 y 3).

Tabla 11: Regresión Logística + cross validation + métodos de rebalanceo de clases (GMSC)

	GMSC_V1			GMSC_V2			GMSC_V3			GMSC_V4			GMSC_V5		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
Log Reg +cv + OVUN	0,815	0,756	0,332	0,765	0,761	0,304	0,765	0,750	0,308	0,713	0,750	0,268	0,738	0,766	0,288
Log Reg +cv + OVER	0,824	0,752	0,337	0,770	0,770	0,313	0,791	0,756	0,328	0,745	0,768	0,293	0,735	0,766	0,287
Log Reg +cv + UNDER	0,821	0,755	0,337	0,828	0,752	0,345	0,759	0,751	0,305	0,782	0,758	0,308	0,822	0,767	0,344
Log Reg +cv + SMOTE	0,910	0,723	0,430	0,916	0,694	0,414	0,913	0,707	0,434	0,905	0,725	0,420	0,898	0,695	0,375

Los modelos “Log Reg” con rebalanceo de clases muestran en todas las versiones un incremento en las métricas de AUC y F1, junto una disminución en la métrica accuracy. Estos resultados muestran claramente que cuando se incorporan los métodos de rebalanceo de clases se produce un trade-off entre accuracy y AUC/F1. Los modelos rebalanceados aprenden más sobre los casos minoritarios (*SeriousDlqin2yrs = 0*) logrando predecir mejor estos casos a costa de una menor cantidad de aciertos totales.

En relación a las distintas metodologías de rebalanceo de clases no se visualiza un método dominante sobre las métricas AUC. Si llama la atención que el método SMOTE obtiene las mejores métricas de F1 con la menor pérdida en términos de accuracy para todas las versiones. Sin embargo dicho método es el de peor performance en términos de AUC comparado con el resto de los métodos de rebalanceo en cada versión.

Tabla 12: Random Forest + métodos de rebalanceo de clases (GMSC)

	GMSC_V1			GMSC_V2			GMSC_V3			GMSC_V4			GMSC_V5		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
R.FOREST	0,935	0,521	0,082	0,935	0,521	0,116	0,932	0,530	0,114	0,936	0,541	0,152	0,934	0,500	NaN(*)
R FOREST + OVUN	0,660	0,750	0,252	0,676	0,761	0,266	0,664	0,752	0,267	0,714	0,765	0,277	0,720	0,771	0,283
R FOREST + OVER	0,659	0,750	0,251	0,659	0,754	0,257	0,665	0,751	0,266	0,714	0,765	0,277	0,733	0,774	0,290
R FOREST + UNDER	0,684	0,757	0,262	0,667	0,758	0,261	0,666	0,753	0,267	0,714	0,765	0,277	0,715	0,772	0,281
R FOREST + SMOTE	0,856	0,762	0,378	0,855	0,756	0,389	0,864	0,748	0,392	0,853	0,767	0,377	0,852	0,761	0,371

Los modelos Random Forest muestran conclusiones similares a las expuestas por los modelos de regresión logística. Los modelos sin rebalanceo muestran performance muy bajas en AUC y F1 para todas las versiones a costa de una alta performance en accuracy. Sin embargo, en todas las versiones excepto la versión 5 las métricas de AUC y F1 son mejores en los modelos Random Forest sin rebalanceo que en los de regresión logística sin rebalanceo. El modelo Random Forest sin rebalanceo aplicado a la versión 5 tiene valor “NaN”

(“Not a Number”) en F1 mostrando que el clasificador predice todas las observaciones como la clase mayoritaria por lo que obtiene 0,5 de AUC.

Observando los modelos Random Forest con rebalanceo se puede concluir que ningún método de rebalanceo resulta dominante en todas las versiones si se observa AUC. También en estos ejemplos es llamativo como SMOTE tiene la menor pérdida en accuracy y los mejores resultados F1 en comparación con el resto de los métodos de rebalanceo en cada versión.

Si se comparan los resultados de los modelos tradicionales versus los modelos ensamblados solamente se puede observar una dominancia en los modelos sin rebalanceo, en los que Random Forest obtiene mejores resultados de AUC y F1 por sobre la regresión logística. Sin embargo, esto no se presenta en los modelos con rebalanceo de clases en los que podemos observar que Random Forest con Over Sampling obtiene la métrica más alta de AUC en la versión 5 pero la métrica más alta de F1 se obtiene en una regresión logística con rebalanceo SMOTE en la versión 3. Los tratamientos que serán tenidos en cuenta para continuar el análisis sobre otro modelo ensamblado como XGBoost y los modelos cost-sensitive serán la versión 5 y la versión 4.

Tabla 13: Regresión Logística + cross validation (HCDR)

	HCDR_V1			HCDR_V2			HCDR_V3			HCDR_V4			HCDR_V5		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
Log Reg	0,922	0,500	0,001	0,922	0,500	0,001	0,922	0,50	NaN	0,920	0,504	0,019	0,920	0,506	0,025
Log Reg +cv	0,686	0,531	0,146	0,686	0,531	0,146	0,348	0,540	0,154	0,920	0,501	0,005	0,914	0,520	0,086

La tabla 13 muestra los resultados de los primeros modelos evaluados sobre la base de datos HCDR. Lo que se puede observar es que al igual que en la base de datos anterior los modelos de regresión logística sin rebalanceo obtienen, en casi todos los casos, buenos resultados de accuracy pero muy bajos resultados de AUC y F1. Esto indica nuevamente que los modelos sin rebalanceo logran una baja tasa de error clasificando correctamente la clase mayoritaria pero incorrectamente la clase minoritaria. Una clara excepción a este comportamiento se puede ver en la regresión logística con validación cruzada de la versión 3 para la cual se obtiene una mejora en las métricas AUC y F1 pero con una disminución muy significativa de accuracy (0,348).

La validación cruzada logró en todos los casos mejorar las métricas AUC y F1 con una reducción en accuracy excepto en el caso de la versión 4 donde el efecto es inverso pero de baja magnitud. El modelo sin validación cruzada implementado en la versión 3 predice en su totalidad la clase mayoritaria obteniendo 0,5 de AUC y “NaN” de F1.

Tabla 14: Regresión Logística + cross validation + métodos de rebalanceo de clases (HCDR)

	HCDR_V1			HCDR_V2			HCDR_V3			HCDR_V4			HCDR_V5		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
Log Reg +cv + OVUN	0,619	0,536	0,151	0,619	0,536	0,151	0,204	0,527	0,150	0,633	0,604	0,198	0,633	0,604	0,198
Log Reg +cv + OVER	0,619	0,536	0,151	0,619	0,536	0,151	0,283	0,545	0,156	0,628	0,603	0,197	0,628	0,603	0,197
Log Reg +cv + UNDER	-	-	-	0,311	0,539	0,154	0,223	0,517	0,147	0,650	0,603	0,200	0,650	0,603	0,200
Log Reg +cv + SMOTE	-	-	-	0,620	0,562	0,167	0,245	0,529	0,151	0,914	0,501	0,018	0,918	0,500	0,006

Cuando se introducen los modelos con rebalanceo de clases se pueden observar algunos resultados de gran interés. La versión 1 muestra una leve mejora en términos de AUC y F1 respecto a los modelos sin rebalanceo. En esta versión se observa que no se obtuvieron resultados para los métodos Under-Sampling y Smote. El método Under-Sampling no obtuvo resultados dado que al eliminar observaciones sobre la base de entrenamiento para rebalancear las clases sobre la variable TARGET se eliminaron por completo niveles que persisten en la base de testeo, por lo que el clasificador presenta errores al momento de predecir resultados. Esto es un problema que puede ser resuelto pero que se pretende exponer para dar noción de los inconvenientes que puede conllevar trabajar con este método en algunas situaciones. En cuanto al método Smote, este no permite trabajar sobre variables categóricas y dado que la versión 1 no comprende tratamientos para convertir variables categóricas en numéricas este método no permite rebalancear la base de entrenamiento.

A nivel general, los resultados de los modelos rebalanceados muestran una mejora en términos de AUC y F1 y una disminución en sus métricas de accuracy. Al comparar los resultados de ambas series de datos se puede observar un trade-off más significativo en HCDR considerando que la tasa de error implícita de los modelos que clasifican todas las observaciones como clase mayoritaria es similar para ambas series pero en esta última las mejoras en términos de AUC y F1 son menores y la pérdida de accuracy es mucho más pronunciada.

La versión 1 logra una leve mejora de AUC y F1. La versión 2 tiene resultados cuasi idénticos a la versión 1 pero logra los mejores resultados con el método Smote. La versión 3 logra superar las métricas AUC y F1 de la versión con validación cruzada con el método Over Sampling (con poca magnitud) pero logra muy bajas métricas en general con muy baja performance en accuracy.

Las versiones 4 y 5 logran los mejores resultados de la muestra, Estos modelos lograron superar 0,6 de AUC y 0,19 de F1 excepto con el método Smote (en ambas versiones) donde se obtuvieron AUC cercanos a 0,5 y F1 cercanos a 0 aunque con bajo trade-off (alto accuracy). Es importante destacar que los niveles de accuracy en estas versiones son más altos que en el resto de las versiones (teniendo en cuenta todos los métodos de rebalanceo) exhibiendo una performance mejor en todas las métricas. Estos mejoras podrían atribuirse a alguna de las diferencias en los tratamientos de estas versiones por sobre el resto (o una combinación de las mismas) entre las que se encuentran modificar la escala de las variables, tratamiento sobre los valores ausentes y la selección de variables (las versiones 4 y 5 implementan la selección a través de una matriz de correlación que elimina variables con mayor correlación a 0,7).

Tabla 15: Random Forest + métodos de rebalanceo de clases (HCDR)

	HCDR_V1			HCDR_V2			HCDR_V3			HCDR_V4			HCDR_V5		
	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1	ACC	AUC	F1
R.FOREST	0,922	0,500	NaN	0,922	0,500	NaN	0,922	0,500	NaN	0,920	0,500	NaN	0,920	0,500	NaN
R FOREST + OVUN	0,515	0,601	0,183	0,552	0,599	0,185	0,256	0,553	0,159	0,633	0,656	0,229	0,657	0,656	0,233
R FOREST + OVER	0,580	0,603	0,189	0,564	0,598	0,185	0,234	0,548	0,157	0,621	0,657	0,228	0,661	0,658	0,235
R FOREST + UNDER	0,511	0,594	0,180	0,535	0,595	0,182	0,223	0,546	0,156	0,656	0,658	0,235	0,665	0,660	0,238
R FOREST + SMOTE	-	-	-	0,762	0,570	0,183	0,347	0,567	0,164	0,920	0,502	0,008	0,920	0,502	0,008

La tabla 15 muestra nuevamente que cuando se implementan los modelos Random Forest se puede observar una mejora considerable en las métricas AUC y F1 en los modelos rebalanceados, obteniendo con alguno de los métodos de rebalanceo la mejor performance para cada versión (AUC/F1). Mientras que los modelos sin rebalanceo obtienen en todos los casos las mismas métricas siendo estos modelos que clasifican a todas las observaciones (o casi todas) como clase mayoritaria ignorando a la clase minoritaria.

En segundo lugar, no se observa un claro método dominante de rebalanceo de clases. Para la versión 1 el método Smote no presenta resultados por el mismo motivo que no presenta resultados aplicado en la regresión logística (problemas al trabajar con variables no numéricas). Los mejores resultados de AUC y F1 se obtienen en las versiones 4 y 5 con el método Under-Sampling siendo esta última versión la de mejor performance. Llama la atención que en estas dos versiones el método Smote presenta resultados similares a los modelos sin rebalanceo (Alto accuracy, bajo AUC/F1).

Los tratamientos que serán tenidos en cuenta para continuar el análisis sobre otro modelo ensamblado como XGBoost y los modelos cost-sensitive serán la versión 5 (por su alto performance en AUC y F1) y la versión 3.

Tabla 16: XGBoost + métodos de rebalanceo de clases (GMSC)

	GMSC_V4			GMSC_V5		
	ACC	AUC	F1	ACC	AUC	F1
XGBoost	0,938	0,861	0,298	0,938	0,859	0,277
XGBoost + OVUN	0,804	0,858	0,333	0,802	0,860	0,334
XGBoost + OVER	0,807	0,860	0,341	0,804	0,857	0,334
XGBoost + UNDER	0,796	0,859	0,330	0,786	0,855	0,321
XGBoost + SMOTE	0,893	0,856	0,412	0,894	0,859	0,416

La tabla 16 muestra los resultados de los modelos XGBoost aplicados a las versiones 4 y 5 de la base de datos GMSC. Estos modelos muestran un incremento significativo en la métrica AUC y resultados dispares para la métrica F1. Se puede observar que el peor resultado de AUC en los modelos XGBoost (0,855 – Under-Sampling V5) supera al mejor resultado de los modelos de regresión logística y Random Forest presentados anteriormente. Mientras que el mejor resultado de la métrica F1 (0,416- Smote V5) es inferior a 3 de los 5 mejores resultados de los modelos de regresión y Random Forest. Los métodos de rebalanceo en estos modelos no muestran una diferencia significativa en el impacto de la métrica AUC aunque sí exhiben diferencias en la métrica F1 para la cual los modelos sin rebalanceo obtienen el resultado más bajo para cada versión.

Tabla 17: XGBoost + cross validation + métodos de rebalanceo de clases (GMSC)

	GMSC_V4			GMSC_V5		
	ACC	AUC	F1	ACC	AUC	F1
XGB GRID.CV	0,939	0,864	0,315	0,938	0,862	0,272
XGB + OVUN GRID.CV	0,792	0,864	0,327	0,800	0,862	0,335
XGB + OVER GRID.CV	0,800	0,865	0,336	0,801	0,862	0,335
XGB + UNDER GRID.CV	0,796	0,863	0,332	0,793	0,861	0,328
XGB + SMOTE GRID.CV	0,890	0,858	0,414	0,893	0,860	0,412

La tabla 17 presenta los resultados de los modelos XGBoost con optimización de hiper parámetros sobre las versiones 4 y 5 de la base de datos GMSC. Los resultados de estos modelos muestran una mejora en la métrica AUC con respecto a los modelos XGBoost sin optimización. Estas mejoras se ubican entre 0,002 (Smote V4) y 0,006 (Over Under - Sampling V4) nominalmente (AUC). En este sentido, si se realiza el mismo análisis sobre la métrica F1 los resultados son dispares ya que en varios casos la métrica mejora con la optimización y en otros empeora aunque siempre lo hacen en poca magnitud.

Una vez más los modelos sin rebalanceo obtuvieron las peores métricas de F1, pero en cuanto a las métricas AUC similar para la versión 4 y la mejor para la versión 5 (junto a Over-Sampling). En relación a los métodos de rebalanceo, Over-Sampling obtuvo los mejores resultados en AUC mientras que Smote obtuvo los mejores resultados en F1.

Tabla 18: XGBoost + métodos de rebalanceo de clases (HCDR)

	HCDR_V3			HCDR_V5		
	ACC	AUC	F1	ACC	AUC	F1
XGB	0,923	0,695	0,005	0,922	0,752	0,032
XGB + OVUN	0,665	0,691	0,220	0,702	0,747	0,258
XGB + OVER	0,663	0,691	0,218	0,704	0,749	0,256
XGB + UNDER	0,642	0,692	0,214	0,686	0,743	0,252
XGB + SMOTE	0,872	0,678	0,220	0,912	0,749	0,209

La tabla 18 muestra los resultados de los modelos XGBoost aplicados a las versiones 3 y 5 de la base de datos HCDR. Al igual que en el caso de la base de datos GMSC se nota un incremento significativo en la métrica AUC. Se puede observar que el peor resultado de AUC en los modelos XGBoost (0,678 – Smote V3) supera al mejor resultado de los modelos de regresión logística y Random Forest presentados anteriormente. En relación a la métrica F1 los resultados son ambiguos. Los modelos XGBoost aplicados a la versión 3 obtienen mejores resultados en la métrica F1 con un incremento significativo comparado con sus pares en los modelos de regresión logística y Random Forest (por método de rebalanceo). Para la versión 5 también hay una mejora en la métrica por método de rebalanceo pero esta mejora es poca significativa.

En cuanto a los métodos de rebalanceo se observa que en los modelos XGBoost el modelo sin rebalanceo obtiene muy buenos resultados para la métrica AUC (el de mejor performance para cada versión) y los peores resultados para la métrica F1. No se observa una tendencia en estos modelos por parte de los modelos rebalanceados. Analizando los resultados para ambas versiones se puede observar una diferencia sustancial en la performance de AUC a favor de la versión 5 (que se mantiene en los modelos de regresión logística y Random Forest para HCDR) y una diferencia, aunque menor, en la performance de F1 a favor de la versión 5 (también representada en los modelos de regresión logística y Random Forest para HCDR).

Tabla 19: XGBoost + cross validation + métodos de rebalanceo de clases (HCDR)

	HCDR_V3			HCDR_V5		
	ACC	AUC	F1	ACC	AUC	F1
XGB GRID.CV	0,923	0,694	0,001	0,921	0,749	0,013
XGB + OVUN GRID.CV	0,663	0,694	0,219	0,688	0,751	0,254
XGB + OVER GRID.CV	0,661	0,694	0,219	0,691	0,751	0,256
XGB + UNDER GRID.CV	0,653	0,694	0,218	0,693	0,749	0,255
XGB + SMOTE GRID.CV	0,874	0,683	0,222	0,916	0,744	0,174

La tabla 19 presenta los resultados de los modelos XGBoost con optimización de hiper parámetros sobre las versiones 3 y 5 de la base de datos HCDR. Los resultados de estos modelos muestran una leve mejora en la métrica AUC con respecto a los modelos XGBoost sin optimización para la mayoría de los casos. Estas mejoras se ubican entre 0,002 (Over Sampling V5) y 0,006 (Under Sampling V5) nominalmente (AUC). Mientras que los modelos sin rebalanceo obtuvieron peores métricas AUC junto con el modelo Smote V5 que tuvo la disminución más significativa (-0,005 nominalmente).

En este sentido, si se realiza el mismo análisis sobre la métrica F1 los resultados son dispares ya que en varios casos la métrica mejora con la optimización y en otros empeora aunque siempre lo hacen en poca magnitud. Nuevamente los modelos sin rebalanceo obtuvieron las peores métricas de F1 para la cual Over-Under Sampling obtuvo la mejor performance (V5).

Llama la atención que para esta serie de datos los mejores resultados sobre la métrica AUC se obtienen en los modelos no optimizados, siendo dicha métrica la elegida en el diseño de los modelos para ser optimizada. Esto sucede dado que la optimización implementada emplea el método de “búsqueda por grilla” en el cual se iteran distintos valores de hiper parámetros y se selecciona la combinación que mayor performance obtiene sobre el proceso de validación cruzada. Siendo este el proceso de optimización se interpreta que los hiper parámetros estáticos elegidos en los modelos sin optimización generaron mejores resultados que la optimización por iteración en estos casos puntuales.

RESULTADOS GENERALES - MODELOS INSENSIBLES AL COSTO

Tomando los resultados anteriores se obtienen las siguientes conclusiones para ambas series de datos:

- En los modelos de regresión logística y Random Forest la ausencia de un método de rebalanceo produce una muy baja performance para las métricas AUC y F1.
- Los modelos de regresión logística sin rebalanceo, con validación cruzada, obtienen mejores métricas que los modelos ensamblados Random Forest sin rebalanceo.
- Los modelos sin rebalanceo, tradicionales o ensamblados, mantienen una tasa de acierto similar a la proporción de la clase mayoritaria.
- No se exhibe un método de rebalanceo dominante aunque el método Smote muestra muy buena performance en la métrica F1.
- Para la base GMSC no se exhibe un tratamiento (versiones) dominante, mientras que para la base HCDR se observa una dominancia por parte de las versiones 4 y 5 en las que se destacan el tratamiento de los valores ausentes, transformación de variables y la selección de variables a través del uso de la matriz de correlaciones.
- Se observa un aumento significativo de la métrica AUC para los modelos XGBoost.
- Los modelos XGBoost sin rebalanceo obtienen performance de AUC similares a las de los modelos rebalanceados siendo también los de peor performance para la métrica F1.

MODELOS COST-SENSITIVE

Cost sensitive (costos fijos - GMSC)

A continuación se presentarán los resultados de los modelos sensibles al costo asumiendo costos fijos para la base de datos GMSC:

Tabla 20: Resultados modelos de regresión logística cost sensitive con costos fijos (GMSC)

	GMSC_V4				GMSC_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
Reg Log	0,934	0,522	0,085	0,319	0,934	0,509	0,036	0,327
Reg Log + th	0,904	0,7180	0,411	0,228	0,934	0,509	0,039	0,327
Reg Log (Smote)	0,812	0,7630	0,333	0,274	-	-	-	-
Reg Log (Smote) + th	0,460	0,681	0,187	0,557	-	-	-	-
Reg Log (Over)	-	-	-	-	0,603	0,629	0,181	0,487
Reg Log (Over) + th	-	-	-	-	0,067	0,500	0,125	0,933

La tabla 20 muestra los modelos de regresión logística, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 4 y 5 de la base de datos GMSC. En esta sección se aplica a cada versión un

método de rebalanceo elegido según los mejores resultados obtenidos anteriormente. En este sentido, a la versión 4 se aplica el método Smote y a la versión 5 se aplica el método Over-Sampling. A su vez, cada modelo es evaluado con el punto de corte en el nivel 0,5 y con el punto de corte teórico ("+th") en 0,83 (tal y como se obtiene del cálculo expresado en la sección de metodología). Se agrega a las métricas de performance la noción de costo promedio.

Los resultados muestran que las regresiones sin rebalanceo obtienen costos promedios similares para ambas versiones. Cuando se modifica el punto de corte asignando el valor teórico la versión 4 logra reducir el costo promedio (aproximadamente en un 30%) mejorando incluso las métricas AUC y F1, mientras que la versión 5 mantiene prácticamente los mismos valores para todas sus métricas.

Cuando se introducen los métodos de rebalanceo los resultados sobre costos son dispares. El método Smote aplicado a la versión 4 muestra un incremento de las métricas AUC y F1 junto a un menor costo promedio si se lo compara con la regresión logística sin rebalanceo pero cuando se modifica el punto de corte se produce una disminución en las métricas AUC y F1 junto a un incremento del 100% aproximadamente del costo promedio. En la versión 5 se observa que el método Over-Sampling mejora las métricas de AUC y F1 con respecto a los modelos sin rebalanceo pero obtiene un costo promedio más elevado (50% aproximadamente), mientras que cuando se modifica el punto de corte se reducen todas las métricas de performance con un incremento del costo promedio en aproximadamente un 100%.

Cuando se modifica el punto de corte en los modelos rebalanceados se observa una caída pronunciada en la métrica accuracy. Esto indica que los modelos aumentan la tasa de error clasificando mayoritariamente la clase minoritaria (*SeriousDlqin2yrs* =1) dado que la probabilidad necesaria para clasificar la clase mayoritaria aumenta de 0,5 a 0,83. En el resultado de la versión 5 se puede observar que la métrica accuracy del modelo de regresión logística sin rebalanceo con punto de corte 0,5 y el modelo de regresión logística Over-Sampling con punto de corte 0,83 son inversamente proporcionales, indicando que el primer modelo clasifica en su mayoría la clase mayoritaria mientras que el último clasifica en su mayoría la clase minoritaria.

Tabla 21: Resultados modelos de XGBoost cost sensitive con costos fijos (GMSC)

	GMSC_V4				GMSC_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
XGB	0,936	0,598	0,303	0,274	0,935	0,587	0,274	0,282
XGB+ th	0,900	0,740	0,424	0,218	0,900	0,728	0,407	0,227
XGB (Smote)	0,898	0,723	0,405	0,289	-	-	-	-
XGB (Smote) + th	0,729	0,770	0,286	0,320	-	-	-	-
XGB (Over)	-	-	-	-	0,839	0,763	0,358	0,247
XGB (Over) + th	-	-	-	-	0,605	0,734	0,229	0,247

La tabla 21 muestra los modelos XGBoost, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 4 y 5 de la base de datos GMSC. Se observa que los primeros modelos XGBoost sin rebalanceo con punto de corte en 0,5 parten de un costo promedio más bajo que los modelos de regresión logística. Ambas versiones al modificar el punto de corte logran mejorar las métricas AUC y F1 mientras que reducen el costo promedio al modificar el punto de corte en los modelos sin rebalanceo.

Nuevamente cuando se introducen los métodos de rebalanceo los resultados son ambiguos. Para la versión 4 el método Smote obtiene un costo promedio más elevado que el modelo sin rebalanceo y la modificación sobre el punto de corte produce un incremento mayor. Para la versión 5 el método Over-Sampling obtiene un costo promedio menor que el modelo sin rebalanceo y la modificación del punto de corte no genera cambios en el costo promedio.

Tabla 22: Resultados modelos de XGBoost + grid search cost sensitive con costos fijos (GMSC)

	GMSC_V4				GMSC_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
XGB + GRID	0,937	0,594	0,295	0,276	0,937	0,584	0,272	0,282
XGB+ GRID + th	0,903	0,745	0,435	0,213	0,902	0,744	0,433	0,214
XGB (Smote) +GRID	0,886	0,715	0,378	0,241	-	-	-	-
XGB (Smote) + GRID + th	0,776	0,756	0,303	0,295	-	-	-	-
XGB (Over) +GRID	-	-	-	-	0,806	0,787	0,344	0,256
XGB (Over) +GRID + th	-	-	-	-	0,422	0,670	0,180	0,589

La tabla 22 muestra los modelos XGBoost optimizando la métrica AUC a través de la búsqueda de hiper parámetros, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 4 y 5 de la base de datos GMSC. Una vez más se observa que los modelos sin rebalanceo

obtienen costos similares a los modelos anteriores (sin rebalanceo). Cuando se altera el punto de corte para los modelos sin rebalanceo se obtiene un incremento en las métricas AUC y F1 mientras que se logra reducir el costo promedio. Sin embargo, los modelos rebalanceados al modificar el punto de corte obtienen costos promedios mayores aunque con el punto de corte en 0,5 logran reducir los costos en comparación a los modelos sin rebalanceo con punto de corte en 0,5.

Los costos promedios más bajos se obtienen, para ambas versiones, con los modelos XGBoost optimizados sin rebalanceo con el punto de corte teórico (0,83).

Cost sensitive (costos fijos - HCDR)

A continuación se presentarán los resultados de los modelos sensibles al costo asumiendo costos fijos para la base de datos HCDR:

Tabla 23: Resultados modelos de regresión logística cost sensitive con costos fijos (HCDR)

	HCDR_V3				HCDR_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
Reg Log	0,922	0,500	NaN	0,388	0,920	0,510	0,043	0,392
Reg Log + th	0,884	0,564	0,198	0,369	0,881	0,608	0,276	0,348
Reg Log (Smote)	0,763	0,619	0,227	0,408	-	-	-	-
Reg Log (Smote) + th	0,110	0,514	0,147	0,893	-	-	-	-
Reg Log (Under)	-	-	-	-	0,685	0,674	0,251	0,423
Reg Log (Under) + th	-	-	-	-	0,168	0,540	0,159	0,837

La tabla 23 muestra los modelos de regresión logística, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 3 y 5 de la base de datos HCDR. En la versión 3 se aplica el método Smote y en la versión 5 se aplica el método Under-Sampling. A su vez, cada modelo es evaluado con el punto de corte en el nivel 0,5 y con el punto de corte teórico (“+th”) en 0,83 (tal y como se obtiene del cálculo expresado en la sección de metodología). Se agrega a las métricas de performance la noción de costo promedio.

Tal como se observa en los modelos GMSC de la sección anterior, los modelos sin rebalanceo presentan altas métricas de accuracy en sintonía con la alta proporción de clase mayoritaria en los datos junto con bajas métricas de AUC y F1. Cuando se altera el punto de corte en estos modelos se mejoran las

métricas de AUC y F1 mientras que se logra reducir levemente el costo promedio en ambas versiones.

Los métodos de rebalanceo obtienen mejores métricas de AUC y F1 pero con costos promedios más elevados que los modelos sin rebalanceo. Cuando se modifica el punto de corte en los modelos rebalanceados estos empeoran en términos de costos en aproximadamente un 100%.

Tabla 24: Resultados modelos de XGBoost cost sensitive con costos fijos (HCDR)

	HCDR_V3				HCDR_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
XGB	0,922	0,502	0,011	0,386	0,919	0,519	0,076	0,387
XGB+ th	0,873	0,581	0,224	0,364	0,857	0,634	0,292	0,344
XGB (Smote)	0,882	0,565	0,199	0,369	-	-	-	-
XGB (Smote) + th	0,473	0,612	0,186	0,596	-	-	-	-
XGB (Under)	-	-	-	-	0,675	0,670	0,246	0,432
XGB (Under) + th	-	-	-	-	0,265	0,577	0,171	0,752

La tabla 24 muestra los modelos XGBoost, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 3 y 5 de la base de datos HCDR. Se observa que los primeros modelos XGBoost sin rebalanceo con punto de corte 0,5 parten de un costo promedio levemente más bajo que los modelos de regresión logística. Ambas versiones al modificar el punto de corte logran mejorar las métricas AUC y F1 mientras que reducen el costo promedio al modificar el punto de corte en los modelos sin rebalanceo.

Los métodos de rebalanceo muestran resultados ambiguos. En la versión 3 el método Smote mejora las métricas AUC y F1 con una leve reducción sobre el costo promedio mientras que cuando se altera el punto de corte se observa un incremento del 60% aproximadamente. En la versión 5 se observa también un incremento en las métricas AUC y F1 cuando se implementa el método Under-Sampling pero con un costo promedio mayor al modelo sin rebalanceo mientras que alterando el punto de corte el costo promedio aumenta en un 75% aproximadamente.

Tabla 25: Resultados modelos de XGBoost + grid search cost sensitive con costos fijos (HCDR)

	HCDR_V3				HCDR_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
XGB + GRID	0,922	0,501	0,002	0,387	0,919	0,515	0,064	0,389
XGB+ GRID + th	0,887	0,577	0,223	0,358	0,859	0,636	0,296	0,342
XGB (Smote) +GRID	0,877	0,576	0,218	0,364	-	-	-	-
XGB (Smote) + GRID + th	0,307	0,580	0,168	0,723	-	-	-	-
XGB Under) +GRID	-	-	-	-	0,688	0,681	0,256	0,416
XGB (Under) +GRID + th	-	-	-	-	0,213	0,560	0,165	0,796

La tabla 25 muestra los modelos XGBoost optimizando la métrica AUC a través de la búsqueda de hiper parámetros, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 3 y 5 de la base de datos HCDR.

Estos resultados se comportan de acuerdo a la secuencia de resultados anteriores. Se observa que los modelos sin rebalanceo obtienen costos similares, una baja tasa de error en relación a la proporción de la clase minoritaria, una baja performance en AUC y F1. Cuando se modifica el punto de corte sobre los modelos sin rebalanceo se logra reducir los costos promedio en ambas versiones. Los métodos de rebalanceo mejoran las métricas AUC y F1 considerablemente, con una reducción en los costos para la versión 3 y un aumento en la versión 5. Finalmente, cuando se altera el punto de corte en los modelos rebalanceados se obtiene un incremento cercano al 100% en los costos promedios para ambas versiones.

Al analizar la totalidad de los resultados obtenidos por los modelos bajo el supuesto de que los costos para cada escenario de clasificación son fijos se pueden trazar algunas conclusiones:

- Los modelos sin rebalanceo con el punto de corte en 0,5 obtienen métricas de performance bajas de AUC y F1 con costos promedios asociados a que identifican correctamente a los individuos que no presentan dificultades de repago pero incorrectamente a los que sí las presentan.
- Cuando se modifica el punto de corte asignando el valor teórico óptimo (0,83) los modelos sin rebalanceo mejoran las métricas de performance AUC y F1. Los resultados indican que en todos los casos se produce una reducción en el costo promedio.

- Los métodos de rebalanceo no muestran una mejora consistente en términos de costos al utilizar modelos cost-sensitive con el enfoque de costos fijos. Más aún, cuando se asigna el punto de corte teórico en 0,83 todos estos modelos sufren de un incremento en los costos considerable que puede superar en algunos casos el 100% de aumento.
- Los menores costos promedios fueron obtenidos por los modelos XGBoost optimizados sin rebalanceo con el punto de corte óptimo en 0,83 para ambas series de datos.

Cost sensitive (costos variables por observación - GMSC)

A continuación se presentarán los resultados de los modelos sensibles al costo asumiendo costos variables por observación para la base de datos GMSC:

Tabla 26: Resultados modelos cost sensitive example dependent (GMSC)

	GMSC_V4				GMSC_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
Reg Log	0,719	0,771	0,282	69,84	0,537	0,688	0,199	86,86
Reg Log (Smote)	0,071	0,502	0,125	123,1	-	-	-	-
Reg Log (Over)	-	-	-	-	0,067	0,500	0,125	124,5
XGB	0,772	0,734	0,287	102,1	0,776	0,742	0,289	100,2
XGB (Smote)	0,479	0,664	0,183	91,98	-	-	-	-
XGB (Over)	-	-	-	-	0,478	0,633	0,171	115,72

La tabla 26 muestra los modelos de regresión logística y XGBoost, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 4 y 5 de la base de datos GMSC. En esta sección se respeta la metodología de aplicar a cada versión un método de rebalanceo elegido según los mejores resultados obtenidos anteriormente. En este sentido, a la versión 4 se aplica el método Smote y a la versión 5 se aplica el método Over-Sampling. Al suponer costos variables por observaciones la métrica de costo promedio cambia de escala en relación a la misma métrica para los modelos que suponen costos fijos.

Tal y como se menciona en la sección de “Modelos”, la tabla 26 muestra resultados de costos promedios en una escalada significativamente distinta a la de los modelos cost sensitive que asumen costos fijos. Esto es principalmente porque los costos fijos generan un costo promedio que puede encontrarse en el rango [0-5] mientras que los modelos que asumen costos variables dependen de *MonthlyIncome* (en el caso de GMSC) y de Loan (en el caso de HCDR). Ambas variables se encuentran en escalas muy distintas pero los resultados no son comparables entre los distintos enfoques de costos.

Teniendo en cuenta que la matriz de costos variables implícita en estos modelos no permite imputar costos negativos en los casos verdaderos negativos (VN – aceptar un crédito a un buen pagador) mencionado en la sección metodológica, esta matriz presenta la misma asimetría en los errores de clasificación que se presenta en la matriz de costos fijos por lo que podrían esperarse ciertas similitudes en los resultados.

Dicho esto, los resultados de la tabla 26 muestran que los modelos de menores costos en ambas versiones son producidos por los modelos sin rebalanceo. En este caso puntual, los modelos de menores costos promedios son los modelos de regresión logística sin rebalanceo. Cuando se introducen los modelos de regresión logística con rebalanceo estos producen un aumento en términos de costos de un 76% para la versión 4 y un 43% para la versión 5. A diferencia de los modelos que asumen costos fijos, los resultados muestran que los modelos XGBoost sin rebalanceo obtienen costos mayores a los modelos de regresión logística sin rebalanceo. En la versión 4, el método Smote aplicado al modelo XGBoost reduce el costo promedio en relación al XGBoost sin rebalanceo mientras que en la versión 5 el método Over-Sampling aplicado modelo XGBoost aumenta el costo promedio en relación al XGBoost sin rebalanceo.

Cost sensitive (costos variables por observación - HCDR)

Tabla 27: Resultados modelos cost sensitive example dependent (HCDR)

	HCDR_V3				HCDR_V5			
	ACC	AUC	F1	Costo (Prom.)	ACC	AUC	F1	Costo (Prom.)
Reg Log	0,217	0,550	0,157	21,75	0,398	0,623	0,191	19,65
Reg Log (Smote)	0,007	0,500	0,144	22,68	-	-	-	-
Reg Log (Under)	-	-	-	-	0,081	0,501	0,148	18,9
XGB	0,452	0,598	0,179	24,09	0,570	0,654	0,219	22,86
XGB (Smote)	0,219	0,532	0,152	23,59	-	-	-	-
XGB (Under)	-	-	-	-	0,153	0,524	0,236	24,47

La tabla 27 muestra los modelos de regresión logística y XGBoost, con y sin rebalanceo, orientados a la minimización del costo promedio aplicados a las versiones 3 y 5 de la base de datos HCDR. Nuevamente se respeta la metodología de aplicar a cada versión un método de rebalanceo elegido según los mejores resultados obtenidos anteriormente. En este sentido, a la versión 3 se aplica el método Smote y a la versión 5 se aplica el método Under-Sampling. Al suponer costos variables por observaciones la métrica de costo promedio cambia de escala en relación a la misma métrica para los modelos que suponen costos fijos.

Para esta serie de datos se observa que los modelos de regresión logística, con y sin rebalanceo, obtienen costos promedios menores a los que obtienen los modelos XGBoost. En el caso de la versión 3, el método Smote obtiene un costo promedio levemente mayor al modelo sin rebalanceo mientras que en la versión 5 el método Under-Sampling obtiene un costo promedio levemente menor al obtenido por el modelo sin rebalanceo (siendo el de menor costo promedio exhibido en la tabla 27).

Un comportamiento relevante se desprende de estos últimos resultados. Los modelos que obtienen los menores costos promedios para cada una de las versiones obtienen métricas accuracy llamativamente bajas (0,217 y 0,081 respectivamente). Esto indica que los modelos que logran mejores costos para esta serie de datos predicen mayoritariamente que los solicitantes no estarían en condiciones de repagar sus préstamos. Esto no sucede para la serie de datos GMSC donde los modelos de menores costos obtienen métricas accuracy mayores a 0,5 (0,7192 y 0,5372 respectivamente). Sin embargo, dado que los modelos no imputan costos negativos para los casos verdaderos negativos (VN) sería prematuro inferir sobre el trade-off entre la tasa de error y el costo promedio. En los siguientes modelos se analizará este comportamiento en detalle.

Cost sensitive (costos variables por observación + optimización del threshold - GMSC)

A continuación se presentarán los resultados de los modelos sensibles al costo y optimizando el punto de corte asumiendo costos variables por observación para la base de datos GMSC:

Tabla 28: Resultados modelos cost sensitive example dependent + optimización del punto de corte (GMSC)

	GMSC_V4					GMSC_V5				
	Costo (prom.)	Th_Costo	Th_ACC	Costo (Costo)	Costo (ACC.)	Costo (prom.)	Th_Costo	Th_ACC	Costo (Costo)	Costo (ACC.)
Reg Log	70,72	0,93	0,53	-14,92	68,05	74,38	0,93	0,72	52,17	74,36
Reg Log (Smote)	-13,81	0,66	0,09	-14,95	67,41	-	-	-	-	-
Reg Log (Over)	-	-	-	-	-	42,79	0,52	0,26	42,25	73,31
XGB	41,24	0,94	0,39	-24,36	53,39	44,84	0,94	0,43	-20,85	52,48
XGB (Smote)	-0,49	0,78	0,1	-25,81	58,03	-	-	-	-	-
XGB (Over)	-	-	-	-	-	-11,53	0,54	0,07	-14,11	50,89
XGB +GRID	44,27	0,93	0,46	-31,72	50,03	47,64	0,93	0,5	-32,05	47,64
XGB (Smote) +GRID	-4,28	0,79	0,02	-16,57	58,75	-	-	-	-	-
XGB (Over) +GRID	-	-	-	-	-	-29,96	0,5	0,06	-29,96	43,73

La tabla 28 muestra los modelos de regresión logística y XGBoost, con y sin rebalanceo, optimizando el punto de corte en relación a la métricas de costo promedio y accuracy aplicados a las versiones 4 y 5 de la base de datos GMSC. Nuevamente se respeta la metodología de aplicar a cada versión un método de rebalanceo elegido según los mejores resultados obtenidos anteriormente. En este sentido, a la versión 4 se aplica el método Smote y a la versión 5 se aplica el método Over-Sampling. Estos modelos suponen costos variables por observación y asumen que el costo de clasificar correctamente a los solicitantes en condiciones de repago es negativo.

Los resultados se leen de la siguiente manera. Para la versión 4, el modelo de regresión logística que supone costos fijos (costo nulo para los casos verdaderos negativos) y optimiza los costos promedios obtiene un costo promedio de 70,72 con el punto de corte en 0,5 si se aplicara la nueva matriz de costos variables que incluyen los costos negativos para los casos verdaderos negativos (VN). En relación a este modelo, el punto de corte óptimo que minimiza el costo se encuentra en 0,93 (se necesita una probabilidad mayor al 93% para clasificar una observación como un buen pagador) con un costo promedio resultante de -14,92. Por otro lado, para el mismo modelo el punto de corte óptimo que maximiza la métrica accuracy (o minimiza la tasa de error) se encuentra en 0,53 (se necesita una probabilidad mayor al 53% para clasificar una observación como un buen pagador) con un resultado de 68,05 de costo promedio.

Estos resultados arrojan conclusiones de gran importancia. En primer lugar se observa que los modelos cost sensitive que optimizan en relación a una matriz de costos fijos, cuando se recalcula la métrica de costo promedio considerando la nueva matriz de costos variables los modelos rebalanceados obtienen, en todos los casos, una mejora por sobre los modelos sin rebalanceo (ej. Costo promedio “Reg Log V4” [70,72] versus costo promedio “Reg Log Smote V4” [-13,81]). En la mayoría de los casos al introducir los métodos de rebalanceo el costo promedio pasa de ser positivo a negativo exceptuando el caso de la regresión logística en la versión 5 donde el método Over-Sampling no alcanza un costo promedio negativo pero logra reducirlo en un 43% aproximadamente.

En relación al punto de corte que minimiza el costo promedio, los resultados muestran que para los modelos sin rebalanceo este valor se encuentra en 0,93 y 0,94. Esto significa que la decisión óptima para el prestamista es otorgar un crédito si el modelo considera que la probabilidad de repago por parte del solicitante es mayor al 93 - 94%. Para los modelos rebalanceados el punto de corte que minimiza el costo promedio disminuye y se encuentran entre 0,5 y 0,79.

Así como los modelos rebalanceados exhiben buenos resultados en términos de costos promedios cuando se considera la nueva matriz de costos variables (“Costo (prom.)”) el mayor impacto al optimizar el punto de corte se produce en los modelos sin rebalanceo. Como ejemplo se puede observar el modelo XGBoost optimizado sin rebalanceo de la versión 4. Este modelo obtiene un costo promedio de 44,27 mientras que la minimización a través del punto de corte logra reducirlo a -31,72 (-172% aproximadamente). En cambio si se observa el comportamiento del XGBoost optimizado con rebalanceo Over-Sampling de la versión 5 el modelo obtiene un costo promedio de -29,96 sin la necesidad de alterar el punto de corte ya que el óptimo se encuentra en 0,5. Dicho esto, es importante destacar que los menores costos son obtenidos, en ambas versiones, por los modelos XGBoost optimizados sin rebalanceo. Para la versión 4, el modelo XGBoost optimizado sin rebalanceo obtiene costo promedio del doble aproximadamente que el mismo modelo con rebalanceo Smote. En otras palabras, el modelo está generando una ganancia del 100% aproximadamente sobre el mismo modelo con rebalanceo. En la versión 5, el modelo XGBoost optimizado sin rebalanceo obtiene un costo mínimo 7% menor al del mismo modelo con rebalanceo Over-Sampling.

La columna “Costo (Acc)” muestra que los modelos al modificar el punto de corte con el objetivo de maximizar accuracy obtienen costos promedios positivos en cada uno de los modelos. Comparando los costos mínimos ((“Costo (Costo)”) versus los costos maximizando accuracy (“Costo (Acc)”) se obtiene una reducción en los costos en el rango entre 30% y 169% siendo el promedio de los resultados una reducción del 125% para esta serie de modelos.

Cost sensitive (costos variables por observación + optimización del threshold - HCDR)

A continuación se presentarán los resultados de los modelos sensibles al costo y optimizando el punto de corte asumiendo costos variables por observación para la base de datos HCDR:

Tabla 29: Resultados modelos cost sensitive example dependent + optimización del punto de corte (HCDR)

	HCDR_V3					HCDR_V5				
	Costo (prom.)	Th_Costo	Th_ACC	Costo (Costo)	Costo (ACC.)	Costo (prom.)	Th_Costo	Th_ACC	Costo (Costo)	Costo (ACC.)
Reg Log	22.251,01	0,93	0,62	10.402,83	22.519,21	21.225,97	0,88	0,22	10.514,91	23.945,77
Reg Log (Smote)	12.203,54	0,57	0,12	10.368,83	22.552,01	-	-	-	-	-
Reg Log (Under)	-	-	-	-	-	8.396,21	0,56	0,05	7.454,30	24.068,44
XGB	22.404,04	0,93	0,17	10.395,58	22.546,53	22.560,35	0,94	0,24	7.211,16	23.937,69
XGB (Smote)	17.115,15	0,76	0,05	10.819,33	22.532,92	-	-	-	-	-
XGB (Under)	-	-	-	-	-	7.824,48	0,48	0,01	7.635,00	24.039,75
XGB +GRID	22.495,9	0,93	0,45	9.482,4	22.506,29	22.645,7	0,93	0,31	7.174,95	23.867,6
XGB (Smote) +GRID	16.452,16	0,73	0,2	9.932,78	22.454,3	-	-	-	-	-
XGB (Under) +GRID	-	-	-	-	-	7.761,98	0,53	0,01	7.225,98	24.102,36

La tabla 29 muestra los modelos de regresión logística y XGBoost, con y sin rebalanceo, optimizando el punto de corte en relación a la métricas de costo promedio y accuracy aplicados a las versiones 3 y 5 de la base de datos HCDR. Nuevamente se respeta la metodología de aplicar a cada versión un método de rebalanceo elegido según los mejores resultados obtenidos anteriormente. En este sentido, a la versión 3 se aplica el método Smote y a la versión 5 se aplica el método Under-Sampling. Estos modelos suponen costos variables por observación y asumen que el costo de clasificar correctamente a los solicitantes en condiciones de repago es negativo.

Los resultados se leen de la siguiente manera. Para la versión 3, el modelo de regresión logística que supone costos fijos (costo nulo para los casos verdaderos negativos) y optimiza los costos promedios obtiene un costo promedio de 22.251,01 con el punto de corte en 0,5 si se aplicara la nueva matriz de costos variables que incluyen los costos negativos para los casos verdaderos negativos (VN). En relación a este modelo, el punto de corte óptimo que minimiza el costo se encuentra en 0,93 (se necesita una probabilidad mayor al 93% para clasificar una observación como un buen pagador) con un costo promedio resultante de 10.402,83. Por otro lado, para el mismo modelo el punto de corte óptimo que maximiza la métrica accuracy (o minimiza la tasa de error) se encuentra en 0,62 (se necesita una probabilidad mayor al 62% para clasificar una observación como un buen pagador) con un resultado de 22.519,21 de costo promedio.

Al igual que en la serie de datos GMSC cuando se aplica la nueva matriz de costos variables sobre los modelos cost sensitive rebalanceados que optimizan

costos fijos obtienen, en todos los casos, una mejora por sobre los modelos sin rebalanceo (ej. Costo promedio “Reg Log V3” [22.251,01] versus costo promedio “Reg Log Smote V3” [12.203,54]). Esta reducción en los costos promedios al introducir un método de rebalanceo sucede en un rango entre 24% y 66%.

En relación al punto de corte que minimiza el costo promedio, los resultados muestran que para los modelos sin rebalanceo este valor se encuentra en 0,93 y 0,94 (exceptuando el caso del modelo de regresión logística sin rebalanceo para la versión 5 donde el punto de corte óptimo es de 0,88). Mientras que para los modelos donde se introduce un método de rebalanceo el punto de corte se ubica en el rango entre 0,48 y 0,76, siendo 0,53 el punto de corte óptimo del modelo que obtiene el menor costo promedio entre los modelos rebalanceados (“XGB Under + Grid” en la versión 5).

Dado que el punto de corte óptimo que minimiza el costo promedio en los modelos no rebalanceados se encuentra más alejado de 0,5 que en los modelos rebalanceados es esperable que el mayor impacto al asignar el punto de corte óptimo sea en los modelos sin rebalanceo. Como ejemplo se puede observar el modelo XGBoost optimizado sin rebalanceo de la versión 5. Este modelo obtiene un costo promedio de 22.645,7 mientras que la minimización a través del punto de corte logra reducirlo a 7.174,95 (-68% aproximadamente). En cambio si se observa el comportamiento del XGBoost con rebalanceo Under-Sampling de la versión 5 el modelo obtiene un costo promedio de 7.824,48 mientras que el valor mínimo posible es de 7.635, apenas un 2% menor. Al igual que en la serie de datos GMSC, los menores costos son obtenidos, en ambas versiones, por los modelos XGBoost optimizados sin rebalanceo. Dicho esto, los modelos XGBoost optimizados con rebalanceo obtienen costos similares. Los costos en la versión 3 a partir del rebalanceo Smote son un 5% más elevados mientras que para la versión 5 el método Under-Sampling obtiene un incremento del 1%.

Para entender el trade-off que se produce entre maximizar la métrica accuracy y minimizar el costo promedio es necesario analizar las columnas “Costo (Costo)” y “Costo (Acc)” en cada uno de los modelos. Este trade-off que exhiben los modelos HCDR muestran una reducción de costos mínima de 52% (XGB Smote V3) y máxima del 70% (XGB + GRID V5) con un promedio ubicado en 61%.

A diferencia de lo exhibido por los modelos GMSC, los modelos HCDR muestran una diferencia considerable entre versiones para la métrica de costos mínimos. Mientras que la versión 5 obtiene un costo promedio de 7.174,95 en el modelo XGBoost optimizado sin rebalanceo, la versión 3 obtiene como

mínimo un costo promedio de 9.482,4 en el modelo XGBoost optimizado sin rebalanceo (32% mayor al de la versión 5).

RESULTADOS GENERALES - COST-SENSITIVE

Tomando los resultados anteriores se obtienen las siguientes conclusiones para ambas series de datos:

- Cuando los modelos buscan minimizar costos considerando una matriz de costos fijos los modelos con rebalanceo obtienen costos mayores a los modelos sin rebalanceo. En estos modelos el punto de corte teórico (0,83) logra reducir los costos para los modelos sin rebalanceo mientras que para los modelos con rebalanceo el punto de corte teórico genera un aumento en los costos. En todos los casos los menores costos son obtenidos por los modelos XGBoost optimizados sin rebalanceo.
- Los modelos que buscan minimizar costos considerando una matriz de costos variables por observación muestran que los modelos de regresión logística logran menores costos que los modelos XGBoost. La introducción de los métodos de rebalanceo no muestran una incidencia clara y consistente en estos casos.
- Cuando se aplican los costos variables por observación y la posibilidad de imputar costos negativos a los casos verdaderos negativos (clasificar correctamente a un buen pagador) los modelos cost-sensitive que minimizan costos fijos muestran que los métodos de rebalanceo tienen un gran impacto reduciendo los costos en todos los casos exhibidos.
- Aplicando la nueva matriz de costos variables por observación los modelos que pueden obtener los menores costos promedio son los XGBoost optimizados sin rebalanceo en todos los casos para los cuales el punto de corte óptimo se encuentra entre 0,93 y 0,94.
- El trade-off entre minimizar los costos y maximizar la métrica accuracy que se exhibe para los modelos en ambas series de datos muestra una diferencia en el rango de 30% y 169% con un promedio general de 93%.

Conclusiones

En base a lo exhibido a lo largo de todo el presente trabajo se pueden observar múltiples resultados por parte de los modelos de machine learning que brindan información valiosa para los agentes encargados de aceptar o rechazar una solicitud crediticia.

Utilizando distintos tratamientos sobre los datos y dos fuentes de datos distintas se demostró que el impacto por parte de distintos tratamientos puede tener un efecto profundo sobre las métricas de performance AUC y F1. Mientras que para la serie de datos GMSC esto no se corrobora, para la serie de datos HCDR los tratamientos muestran un diferencial a partir de componentes como el manejo de los valores ausentes, la transformación de variables y la selección de variables a través del uso de la matriz de correlaciones.

El uso de modelos de machine learning tradicionales y ensamblados muestra que mientras el mejor modelo Random Forest para ambas series de datos obtiene mejores métricas de performance AUC y F1 que el mejor modelo de regresión logística, no se exhibe una dominancia consistente por parte de los modelos Random Forest. Sin embargo, los modelos ensamblados XGBoost si muestran una dominancia ante el resto de los modelos en todos los casos para las métricas mencionadas.

Quienes están a cargo de evaluar el riesgo crediticio de las solicitudes enfrentan dos grandes desafíos: una gran desproporción de clases en los datos y una gran asimetría de costos. Este estudio presenta cientos de modelos de clasificación con varias métricas para intentar entender cuáles son las herramientas más adecuadas a la hora de confeccionar un modelo de riesgo.

Ante el primer desafío, se utilizaron cuatro tipos de métodos de rebalanceo de clases para entender cómo se comportan en relación a los modelos sin rebalanceo y si alguno de estos métodos tiene mejor performance que el resto. Se demostró que los métodos de rebalanceo muestran un impacto positivo en los modelos de regresión logística y Random Forest pero para los modelos XGBoost los métodos de rebalanceo no producen un incremento sustancial (y en muchos casos los modelos sin rebalanceo son mejores).

El segundo desafío hace referencia a una problemática que produjo una gran cantidad de literatura académica. Estableciendo una matriz de costos fijos y variables por observación se demostraron distintos enfoques para definir la función objetivo del problema cuando se desconoce los costos reales asociados a cada institución crediticia.

Tomando algunos supuestos como la tasa de interés y el valor del préstamo (en el caso de la serie de datos GMSC) este trabajo muestra una metodología para adaptar un modelo cost-sensitive que optimiza costos fijos en un modelo que considera costos variables y soporta costos negativos para los casos en los que se acepta un crédito que se encuentra en condiciones de ser repagado. Modificando el punto de corte se expuso el trade-off que existe entre minimizar la tasa de error y minimizar el costo promedio. En este sentido el agente de riesgo crediticio cuenta con evidencia cuantitativa para entender que un modelo orientado a maximizar la métrica accuracy puede estar generando altos costos y que los costos mínimos podrían ser alcanzados sin la necesidad de implementar métodos de rebalanceo de datos.

Como reflexión final se destaca la necesidad de continuar la línea de investigación sobre esta temática. Dado que los costos asociados al otorgamiento de créditos son distintos para cada institución crediticia, es fundamental que los agentes de riesgo crediticio cuenten con modelos y herramientas que sean validadas en escenarios con distintas magnitudes en la asimetría. Dicho esto, este trabajo puede ser un punto de partida para continuar investigando sobre la relación entre el punto de corte óptimo y las métricas de costos, como puede ser la pérdida de eficiencia con respecto al movimiento del punto de corte utilizando distintos enfoques al estimar los costos (es decir, cuánta rentabilidad pierdo si cambio el threshold de decisión). La importancia que tiene la solvencia de las instituciones crediticias sobre el resto de la economía implica una necesidad alrededor de esta temática.

Bibliografía

- N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over- sampling technique,” Journal of artificial intelligence research, 321-357, 2002. [Link]
- A. Correa Bahnsen, D. Aouada, B. Ottersten, “Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring”, International Conference on Machine Learning and Applications, 263-268, 2014 [Link]
- L. Gambacorta, Y. Huang, H. Qiu, J. Wang, “How do machine learning and non-traditional data affect credit scoring? New evidence from a Chinese fintech firm”, Bank for international settlements, 1-28, 2019. [Link]
- S. Lessmann, B. Baesens, H. V. Seon, L. C. Thomas, “Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research”, European Journal of Operational Research, 2015. [Link]
- M. Saidi, N. Settouti, M. E. H. Daho, M. E. A. Bechar, “Comparison of ensemble cost sensitive algorithms: application to credit scoring prediction”, Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering, 56-59, 2018. [Link]
- V. Sheng, C. X. Ling, “Thresholding for Making Classifiers Cost Sensitive”, American Association for Artificial Intelligence, 2006. [Link]

ANEXO

Anexo 1 - Descripción de variables de la base de datos HCDR.

Variable	Descripción
SK_ID_CURR	ID of loan in our sample
TARGET	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)
NAME_CONTRACT_TYPE	Identification if loan is cash or revolving
CODE_GENDER	Gender of the client
FLAG_OWN_CAR	Flag if the client owns a car
FLAG_OWN_REALTY	Flag if client owns a house or flat
CNT_CHILDREN	Number of children the client has
AMT_INCOME_TOTAL	Income of the client
AMT_CREDIT	Credit amount of the loan
AMT_ANNUITY	Loan annuity
AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given
NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan
NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,Ö)
NAME_EDUCATION_TYPE	Level of highest education the client achieved
NAME_FAMILY_STATUS	Family status of the client
NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)
REGION_POPULATION_RELATIVE	Normalized population of region where client lives (higher number means the client lives in more populated region)
DAYS_BIRTH	Client's age in days at the time of application
DAYS_EMPLOYED	How many days before the application the person started current employment
DAYS_REGISTRATION	How many days before the application did client change his registration
DAYS_ID_PUBLISH	How many days before the application did client change the identity document with which he applied for the loan
OWN_CAR_AGE	Age of client's car
FLAG_MOBIL	Did client provide mobile phone (1=YES, 0=NO)
FLAG_EMP_PHONE	Did client provide work phone (1=YES, 0=NO)
FLAG_WORK_PHONE	Did client provide home phone (1=YES, 0=NO)
FLAG_CONT_MOBILE	Was mobile phone reachable (1=YES, 0=NO)
FLAG_PHONE	Did client provide home phone (1=YES, 0=NO)
FLAG_EMAIL	Did client provide email (1=YES, 0=NO)
OCCUPATION_TYPE	What kind of occupation does the client have
CNT_FAM_MEMBERS	How many family members does client have
REGION_RATING_CLIENT	Our rating of the region where client lives (1,2,3)
REGION_RATING_CLIENT_W_CITY	Our rating of the region where client lives with taking city

	into account (1,2,3)
WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for the loan
HOUR_APPR_PROCESS_START	Approximately at what hour did the client apply for the loan
REG_REGION_NOT_LIVE_REGION	Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)
REG_REGION_NOT_WORK_REGION	Flag if client's permanent address does not match work address (1=different, 0=same, at region level)
LIVE_REGION_NOT_WORK_REGION	Flag if client's contact address does not match work address (1=different, 0=same, at region level)
REG_CITY_NOT_LIVE_CITY	Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)
REG_CITY_NOT_WORK_CITY	Flag if client's permanent address does not match work address (1=different, 0=same, at city level)
LIVE_CITY_NOT_WORK_CITY	Flag if client's contact address does not match work address (1=different, 0=same, at city level)
ORGANIZATION_TYPE	Type of organization where client works
EXT_SOURCE_1	Normalized score from external data source
EXT_SOURCE_2	Normalized score from external data source
EXT_SOURCE_3	Normalized score from external data source
APARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
BASEMENTAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
YEARS_BEGINEXPLUATATION_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
YEARS_BUILD_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
COMMONAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
ELEVATORS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
ENTRANCES_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
FLOORSMAX_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMIN_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LANDAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LIVINGAPARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LIVINGAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
NONLIVINGAPARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
NONLIVINGAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
APARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
BASEMENTAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
YEARS_BEGINEXPLUATATION_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
YEARS_BUILD_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
COMMONAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
ELEVATORS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
ENTRANCES_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMAX_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
FLOORSMIN_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LANDAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LIVINGAPARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LIVINGAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
NONLIVINGAPARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
NONLIVINGAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
APARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
BASEMENTAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
YEARS_BEGINEXPLUATATION_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
YEARS_BUILD_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
COMMONAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
ELEVATORS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ENTRANCES_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
FLOORSMAX_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
FLOORSMIN_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LANDAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LIVINGAPARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
LIVINGAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
NONLIVINGAPARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
NONLIVINGAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
FONDKAPREMONT_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
HOUSETYPE_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
TOTALAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
WALLSMATERIAL_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
EMERGENCYSTATE_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

OBS_30_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings with observable 30 DPD (days past due) default
DEF_30_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings defaulted on 30 DPD (days past due)
OBS_60_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings with observable 60 DPD (days past due) default
DEF_60_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings defaulted on 60 (days past due) DPD
DAYS_LAST_PHONE_CHANGE	How many days before application did client change phone
FLAG_DOCUMENT_2	Did client provide document 2
FLAG_DOCUMENT_3	Did client provide document 3
FLAG_DOCUMENT_4	Did client provide document 4
FLAG_DOCUMENT_5	Did client provide document 5
FLAG_DOCUMENT_6	Did client provide document 6
FLAG_DOCUMENT_7	Did client provide document 7
FLAG_DOCUMENT_8	Did client provide document 8
FLAG_DOCUMENT_9	Did client provide document 9
FLAG_DOCUMENT_10	Did client provide document 10
FLAG_DOCUMENT_11	Did client provide document 11
FLAG_DOCUMENT_12	Did client provide document 12
FLAG_DOCUMENT_13	Did client provide document 13
FLAG_DOCUMENT_14	Did client provide document 14
FLAG_DOCUMENT_15	Did client provide document 15
FLAG_DOCUMENT_16	Did client provide document 16
FLAG_DOCUMENT_17	Did client provide document 17
FLAG_DOCUMENT_18	Did client provide document 18
FLAG_DOCUMENT_19	Did client provide document 19
FLAG_DOCUMENT_20	Did client provide document 20
FLAG_DOCUMENT_21	Did client provide document 21
AMT_REQ_CREDIT_BUREAU_HOUR	Number of enquiries to Credit Bureau about the client one hour before application
AMT_REQ_CREDIT_BUREAU_DAY	Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application)
AMT_REQ_CREDIT_BUREAU_WEEK	Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application)
AMT_REQ_CREDIT_BUREAU_MON	Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application)
AMT_REQ_CREDIT_BUREAU_QRT	Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)
AMT_REQ_CREDIT_BUREAU_YEAR	Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)

Anexo 2 - Tabla de correlaciones absolutas mayores a 0,7, HCDR

La siguiente tabla se obtiene aplicando “one hot encoding”, luego reemplazando los valores ausentes por los valores promedios en todas las columnas y finalmente calculando la matriz de correlación. El resultado que se exhibe son los pares con correlación absoluta mayor a 0,7 .

Variable 1	Variable 2	Coeficiente de Correlación
NAME_CONTRACT_TYPECash.loans	NAME_CONTRACT_TYPERevolving.loans	1
CODE_GENDERF	CODE_GENDERM	1
FLAG_OWN_CARN	FLAG_OWN_CARY	1
FLAG_OWN_REALTYN	FLAG_OWN_REALTYY	1
NAME_INCOME_TYPEPensioner	DAYS_EMPLOYED	1
NAME_INCOME_TYPEPensioner	FLAG_EMP_PHONE	1
NAME_INCOME_TYPEPensioner	ORGANIZATION_TYPEXNA	1
DAYS_EMPLOYED	FLAG_EMP_PHONE	1
DAYS_EMPLOYED	ORGANIZATION_TYPEXNA	1
FLAG_EMP_PHONE	ORGANIZATION_TYPEXNA	1
APARTMENTS_AVG	APARTMENTS_MEDI	1
YEARS_BUILD_AVG	YEARS_BUILD_MEDI	1
COMMONAREA_AVG	COMMONAREA_MEDI	1
ELEVATORS_AVG	ELEVATORS_MEDI	1
ENTRANCES_AVG	ENTRANCES_MEDI	1
FLOORSMAX_AVG	FLOORSMAX_MEDI	1
FLOORSMIN_AVG	FLOORSMIN_MEDI	1
LIVINGAREA_AVG	LIVINGAREA_MEDI	1
OBS_30_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE	1
AMT_CREDIT	AMT_GOODS_PRICE	0,99
BASEMENTAREA_AVG	BASEMENTAREA_MEDI	0,99
YEARS_BEGINEXPLUATATION_AVG	YEARS_BEGINEXPLUATATION_MEDI	0,99
YEARS_BUILD_AVG	YEARS_BUILD_MODE	0,99
FLOORSMAX_AVG	FLOORSMAX_MODE	0,99
FLOORSMIN_AVG	FLOORSMIN_MODE	0,99
LANDAREA_AVG	LANDAREA_MEDI	0,99
LIVINGAPARTMENTS_AVG	LIVINGAPARTMENTS_MEDI	0,99
NONLIVINGAPARTMENTS_AVG	NONLIVINGAPARTMENTS_MEDI	0,99
NONLIVINGAREA_AVG	NONLIVINGAREA_MEDI	0,99
YEARS_BUILD_MODE	YEARS_BUILD_MEDI	0,99
FLOORSMAX_MODE	FLOORSMAX_MEDI	0,99
FLOORSMIN_MODE	FLOORSMIN_MEDI	0,99
COMMONAREA_AVG	COMMONAREA_MODE	0,98

ELEVATORS_AVG	ELEVATORS_MODE	0,98
ENTRANCES_AVG	ENTRANCES_MODE	0,98
APARTMENTS_MODE	APARTMENTS_MEDI	0,98
BASEMENTAREA_MODE	BASEMENTAREA_MEDI	0,98
COMMONAREA_MODE	COMMONAREA_MEDI	0,98
ELEVATORS_MODE	ELEVATORS_MEDI	0,98
ENTRANCES_MODE	ENTRANCES_MEDI	0,98
LANDAREA_MODE	LANDAREA_MEDI	0,98
LIVINGAPARTMENTS_MODE	LIVINGAPARTMENTS_MEDI	0,98
NONLIVINGAPARTMENTS_MODE	NONLIVINGAPARTMENTS_MEDI	0,98
NONLIVINGAREA_MODE	NONLIVINGAREA_MEDI	0,98
HOUSETYPE_MODE	HOUSETYPE_MODEblock.of.flats	0,98
EMERGENCYSTATE_MODE	EMERGENCYSTATE_MODENo	0,98
APARTMENTS_AVG	APARTMENTS_MODE	0,97
BASEMENTAREA_AVG	BASEMENTAREA_MODE	0,97
YEARS_BEGINEXPLUATATION_AVG	YEARS_BEGINEXPLUATATION_MODE	0,97
LANDAREA_AVG	LANDAREA_MODE	0,97
LIVINGAPARTMENTS_AVG	LIVINGAPARTMENTS_MODE	0,97
LIVINGAREA_AVG	LIVINGAREA_MODE	0,97
NONLIVINGAPARTMENTS_AVG	NONLIVINGAPARTMENTS_MODE	0,97
NONLIVINGAREA_AVG	NONLIVINGAREA_MODE	0,97
LIVINGAREA_MODE	LIVINGAREA_MEDI	0,97
YEARS_BEGINEXPLUATATION_MODE	YEARS_BEGINEXPLUATATION_MEDI	0,96
HOUSETYPE_MODE	WALLSMATERIAL_MODE	0,96
REGION_RATING_CLIENT	REGION_RATING_CLIENT_W_CITY	0,95
HOUSETYPE_MODE	EMERGENCYSTATE_MODE	0,95
HOUSETYPE_MODEblock.of.flats	WALLSMATERIAL_MODE	0,94
HOUSETYPE_MODE	EMERGENCYSTATE_MODENo	0,93
HOUSETYPE_MODEblock.of.flats	EMERGENCYSTATE_MODE	0,93
WALLSMATERIAL_MODE	EMERGENCYSTATE_MODE	0,93
HOUSETYPE_MODEblock.of.flats	EMERGENCYSTATE_MODENo	0,92
WALLSMATERIAL_MODE	EMERGENCYSTATE_MODENo	0,92
LIVINGAREA_AVG	TOTALAREA_MODE	0,91
LIVINGAREA_MEDI	TOTALAREA_MODE	0,91
NAME_EDUCATION_TYPEHigher.education	NAME_EDUCATION_TYPESecondary...secondary.special	0,89
APARTMENTS_AVG	LIVINGAREA_AVG	0,89
APARTMENTS_AVG	LIVINGAREA_MEDI	0,89
LIVINGAREA_AVG	APARTMENTS_MEDI	0,89
APARTMENTS_MODE	LIVINGAREA_MODE	0,89
LIVINGAREA_MODE	TOTALAREA_MODE	0,89
APARTMENTS_MEDI	LIVINGAREA_MEDI	0,89
CNT_CHILDREN	CNT_FAM_MEMBERS	0,88

APARTMENTS_AVG	TOTALAREA_MODE	0,88
APARTMENTS_AVG	LIVINGAREA_MODE	0,87
LIVINGAREA_AVG	APARTMENTS_MODE	0,87
APARTMENTS_MODE	LIVINGAREA_MEDI	0,87
LIVINGAREA_MODE	APARTMENTS_MEDI	0,87
APARTMENTS_MEDI	TOTALAREA_MODE	0,87
REG_REGION_NOT_WORK_REGION	LIVE_REGION_NOT_WORK_REGION	0,86
DEF_30_CNT_SOCIAL_CIRCLE	DEF_60_CNT_SOCIAL_CIRCLE	0,86
APARTMENTS_MODE	TOTALAREA_MODE	0,85
ELEVATORS_AVG	LIVINGAREA_AVG	0,84
ELEVATORS_AVG	LIVINGAREA_MEDI	0,84
LIVINGAREA_AVG	ELEVATORS_MEDI	0,84
ELEVATORS_MEDI	LIVINGAREA_MEDI	0,84
REG_CITY_NOT_WORK_CITY	LIVE_CITY_NOT_WORK_CITY	0,83
ELEVATORS_MODE	LIVINGAREA_MODE	0,83
ELEVATORS_MODE	LIVINGAREA_MEDI	0,83
FONDKAPREMONT_MODE	FONDKAPREMONT_MODEreg.oper.account	0,83
ELEVATORS_AVG	TOTALAREA_MODE	0,82
LIVINGAREA_AVG	ELEVATORS_MODE	0,82
APARTMENTS_MEDI	ELEVATORS_MEDI	0,82
APARTMENTS_AVG	ELEVATORS_AVG	0,81
APARTMENTS_AVG	ELEVATORS_MEDI	0,81
ELEVATORS_AVG	LIVINGAREA_MODE	0,81
ELEVATORS_AVG	APARTMENTS_MEDI	0,81
LIVINGAREA_MODE	ELEVATORS_MEDI	0,81
ELEVATORS_MEDI	TOTALAREA_MODE	0,81
NAME_TYPE_SUITEFamily	NAME_TYPE_SUITEUnaccompanied	0,8
APARTMENTS_AVG	ELEVATORS_MODE	0,8
APARTMENTS_MODE	ELEVATORS_MODE	0,8
ELEVATORS_MODE	APARTMENTS_MEDI	0,8
ELEVATORS_MODE	TOTALAREA_MODE	0,8
APARTMENTS_MODE	ELEVATORS_MEDI	0,79
ELEVATORS_AVG	APARTMENTS_MODE	0,78
AMT_CREDIT	AMT_ANNUITY	0,77
AMT_ANNUITY	AMT_GOODS_PRICE	0,77
APARTMENTS_AVG	LIVINGAPARTMENTS_AVG	0,76
APARTMENTS_AVG	LIVINGAPARTMENTS_MEDI	0,76
LIVINGAPARTMENTS_AVG	APARTMENTS_MEDI	0,76
APARTMENTS_MODE	LIVINGAPARTMENTS_MODE	0,76
APARTMENTS_MEDI	LIVINGAPARTMENTS_MEDI	0,76
APARTMENTS_AVG	LIVINGAPARTMENTS_MODE	0,75
LIVINGAPARTMENTS_MODE	APARTMENTS_MEDI	0,75

APARTMENTS_MODE	LIVINGAPARTMENTS_MEDI	0,74
LIVINGAPARTMENTS_AVG	APARTMENTS_MODE	0,73
LIVINGAPARTMENTS_AVG	LIVINGAREA_AVG	0,71
LIVINGAPARTMENTS_AVG	LIVINGAREA_MEDI	0,71
LIVINGAREA_AVG	LIVINGAPARTMENTS_MEDI	0,71
LIVINGAPARTMENTS_MODE	LIVINGAREA_MODE	0,71
LIVINGAPARTMENTS_MODE	LIVINGAREA_MEDI	0,71
LIVINGAPARTMENTS_MEDI	LIVINGAREA_MEDI	0,71
LIVINGAREA_AVG	LIVINGAPARTMENTS_MODE	0,7