LICENCIATURA EN ECONOMIA

DETERMINACIÓN DE ALGORITMOS DE CLASIFICACIÓN ÓPTIMOS PARA LA EVALUACIÓN DE RIESGO CREDITICIO

CASO DE ESTUDIO: PYMES QUE OPERAN EN PLATAFORMAS P2P

Por:

Juan Gabriel Garcia Ojeda

Reg.: 31105

Juan.garcia@fce.uncu.edu.ar

Profesor Tutor

Pablo Mahnic

Mendoza 2023

RESUMEN TECNICO

En las últimas décadas, y especialmente en los últimos años, el sector financiero se ha visto empujado a desarrollar nuevas técnicas que sirvan para estimar la probabilidad de default de la manera más eficiente y precisa posible a partir de una mayor demanda de créditos y una coyuntura macroeconómica en algunos periodos cada vez más volátil. En relación con esto, la reciente aparición de nuevos actores dentro del sistema financiero como las *Fintechs* plantean nuevos desafíos debido a la ruptura con el esquema tradicional del sector bancario. Las plataformas P2P son una de las que más relevancia ha tomado en los últimos años dentro del mercado crediticio debido principalmente a la rapidez en el otorgamiento de créditos. Sin embargo, el crecimiento de estas plataformas puede acarrear un significativo riesgo para la estabilidad financiera debido a malos incentivos que aparecen con su funcionamiento, y es a partir de ello que la presente investigación se propone determinar cuál algoritmo de clasificación presenta un mejor desempeño en la obtención de un modelo de puntaje crediticio para los clientes que operan en dichas plataformas.

El presente trabajo consiste en un análisis explicativo y proyectivo en tanto el mismo es de finalidad aplicada. A partir de la base de datos construida en el trabajo de Guidici *et al*. (2019) acerca de empresas PyMES italianas que participan en plataformas P2P para el año 2015, se lleva a cabo un análisis de estadística descriptiva para conocer las características de los datos y posteriormente se utilizan técnicas econométricas y de entrenamiento de algoritmos de aprendizaje automático para la determinación del algoritmo de clasificación óptimo.

Los resultados indican que: 1) A partir de las diversas métricas de rendimiento utilizadas el algoritmo SVM resulta ser el óptimo, y 2) El algoritmo SVM supera en todas las métricas al clásico modelo Logit.

Palabras claves: Aprendizaje automático, Algoritmos de clasificación, *Fintechs*, *Peer to peer lending.*

INDICE

INTRODUCCION Y ANTECEDENTES

Las instituciones financieras han visto incrementada la demanda de créditos a lo largo de las últimas décadas, siendo este aumento explicado en parte por el surgimiento de nuevos modelos de negocios dentro del sistema financiero comúnmente denominados finanzas descentralizadas. Esta situación, en conjunto con las diversas regulaciones que exigen los países con relación a la necesidad de llevar a cabo algún tipo de evaluación previa que determine el riesgo que implica el otorgamiento del crédito a determinado individuo o empresa, han llevado a dichas instituciones a desarrollar diversas técnicas que sirvan para estimar la probabilidad de default de la manera más eficiente y precisa posible. En este sentido, los modelos de puntaje crediticio han sido los más utilizados.

La mayor regulación del sector, y otros factores como la desconfianza generada producto de la crisis financiera internacional originada en el 2008 produjo un impulso a una de las *FinTechs* más populares en la actualidad denominada *Peer to peer lending* (P2P) la cual ha puesto en mayor relieve la necesidad de mejorar las técnicas utilizadas en la estimación de los modelos de puntaje debido a las características propias de estas plataformas, ya que suponen, entre otras cosas, un riesgo sistémico mayor. Además, a medida que crece el volumen operado a través de estas plataformas, una estimación inadecuada del riesgo puede implicar una amenaza para la estabilidad del sistema financiero. Match *et al*. (2014) en su estudio acerca de este tipo de plataformas para el financiamiento de empresas PyMES señala que los montos prestados se han duplicado entre 2007 y 2014 en Estados Unidos. En el caso de Argentina, las estadísticas de la principal plataforma, Afluenta, muestran que el número de préstamos otorgados a la fecha crece mensualmente y que experimentó un crecimiento exponencial entre Julio de 2015 y Marzo de 2020.

En los últimos años se ha visto un avance importante por parte de la industria en el uso de técnicas de *Machine Learning* para el desarrollo de modelos de puntaje crediticio como alternativa al uso de modelos econométricos clásicos, ya que en ciertos contextos los algoritmos de clasificación desarrollados a partir de Inteligencia Artificial han probado tener un mejor desempeño. Por lo tanto, dado el diagnóstico de situación actual en relación con el crecimiento de las plataformas P2P y el problema de la estimación de su riesgo surge la pregunta: ¿Existe un

modelo de puntaje crediticio óptimo para plataformas P2P?, entendido óptimo como aquel que logra minimizar el riesgo de default. En este sentido se encuadra el extensivo estudio acerca del estado del arte de Lessmann *et al*. (2015) en el que se comparan 41 algoritmos con distintos criterios de evaluación de desempeño predictivo en el uso de bases de datos del tipo *retail*[1] cuyos resultados indican el superior desempeño predictivo del algoritmo de Random Forest. Además, el reciente trabajo de Ko *et al*. (2022) en el cual se evalúan modelos predictivos a partir de métodos estadísticos y métodos basados en (AI) utilizando la base de datos provista por la plataforma P2P más importante de Estados Unidos, Lending Club, y donde llegan a la conclusión de que el algoritmo de ensamble *LightGBM* es el óptimo, ofrece un desarrollo robusto en el tratamiento de las clases desbalanceadas y la utilización de varias métricas de desempeño. Sin embargo, estos autores llevan a cabo sus estudios considerando únicamente a los clientes que conforman el universo *retail*, dejando de lado el universo corporativo. En este sentido, el trabajo llevado a cabo por Guidici *et al*. (2019) utiliza datos financieros de pequeñas y medianas empresas para la obtención del modelo óptimo, aunque se utilizan sólo algunas métricas dentro del conjunto considerado por la literatura para la evaluación de estos modelos. Este autor llegaría a la elección del modelo Logit como óptimo.

En base a lo anterior es que en el presente trabajo se busca llevar a cabo un análisis comparativo de los principales algoritmos de clasificación utilizados en la actualidad aplicados a la estimación de modelos de puntaje crediticio para la evaluación del riesgo de empresas PyMES que acceden a financiamiento mediante instituciones no tradicionales como la *FinTech* P2P. Se propone llevar a cabo dicho análisis a través del uso de diversas métricas que permitan obtener conclusiones e identificar ventajas y desventajas en cada uno de los algoritmos, como así también poder comparar el desempeño con modelos econométricos tradicionales como el modelo Logit.

Debido a la naturaleza de los datos en los que se centrará la presente investigación los algoritmos a evaluar serán del tipo de clasificación. Este tipo de algoritmos se encuentran en el universo de lo que se denomina “aprendizaje supervisado”, es decir, estimación de modelos a partir de datos previamente etiquetados hacia alguna clase. En el presente trabajo se utilizan datos clasificados de manera binaria (0 y 1) siendo 1 la clase positiva que indica la presencia de la característica (*default* en el pago de obligaciones) y 0 la ausencia de esta.

[1] Hace referencia al cliente minorista. Utiliza como fuente de datos información sobre formularios de aplicación, datos demográficos y el historial de transacciones.

CAPITULO I
MARCO TEORICO

El presente capítulo se enfoca en el desarrollo del marco teórico relativo al aprendizaje automático de forma general a partir de los trabajos de Ahumada *et al.* (2018), Hastie *et al.* (2009) y James *et al.* (2018). En estos se puede encontrar un desarrollo más profundo del tema.

1. APRENDIZAJE AUTOMATICO

El aprendizaje automático o aprendizaje estadístico es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos matemáticos que permiten aprender a partir de los datos. Es decir, los sistemas de aprendizaje automático tienen la capacidad de mejorar su rendimiento en una tarea a medida que tienen acceso a mayor cantidad de información. Estos algoritmos se basan en técnicas estadísticas y matemáticas para identificar patrones y relaciones en los datos y posteriormente utilizar estos patrones para hacer predicciones o tomar decisiones sobre nuevos datos.

Según Ahumada *et al.* (2018) el objetivo central del aprendizaje automático es predecir correctamente fuera de la muestra, es decir, se busca evaluar la capacidad predictiva del modelo para datos que no fueron los usados para estimarlo. Por lo tanto, el aprendizaje automático es fundamentalmente una estrategia de construcción de modelos, al contrario del enfoque econométrico clásico frecuentista. La principal diferencia radica en que el aprendizaje automático consiste en construir un modelo con fines predictivos sin necesariamente suponer que existe un modelo dado o un proceso que genera los datos. Por ejemplo, para el caso del modelo clásico de regresión lineal predecir una variable y en base a x consiste en estimar $f(x) = E(y|x)$ utilizando una base de datos $(x_i, y_i),\ i = 1, 2, \dots, n$ a partir del modelo

$$y = x\beta + \mu\,, (1.1)$$

donde x es un vector transpuesto de regresores con k componentes y cuya primera componente es igual a 1, mientras que β es un vector cuyas k componentes conforman los coeficientes del modelo

y μ es el término de error. A partir de una muestra de datos $(x_i, y_i), i = 1, 2, \dots, n$ se obtienen los estimadores $\hat{\beta}$ MCO de modo que:

$$\hat{y}_i = x_i\hat{\beta}.$$

La diferencia entre la econometría clásica y el aprendizaje automático es que la primera considera al modelo (1.1) como dado y concentra sus esfuerzos en estimar β de la mejor manera posible, donde mejor hace referencia a propiedades tales como insesgadez y varianza mínima, mientras que, por el contrario, el aprendizaje automático ve a $x\beta$ como una de las posibles configuraciones de una relación genérica $f(x)$, candidatas a predecir correctamente a y fuera de la muestra.

1.1. APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado es una categoría de aprendizaje automático en el que se le proporciona al algoritmo un set de datos sin etiquetas o categorías previas, y a partir de estos datos el algoritmo busca encontrar patrones y estructuras ocultas en los datos sin la necesidad de una supervisión externa. En términos más formales, el aprendizaje no supervisado trata con el problema de encontrar una distribución de probabilidad subyacente que describe los datos.

El objetivo principal del aprendizaje no supervisado es encontrar patrones y estructuras en los datos que sean útiles para una variedad de aplicaciones tales como *clustering*, reducción de dimensionalidad y detección de anomalías. Los algoritmos de *clustering* buscan agrupar los datos en *clusters* basándose en la similitud o distancia entre ellos. Los algoritmos de reducción de dimensionalidad buscan representar los datos en un espacio de menor dimensión manteniendo la mayor cantidad de información posible, y los algoritmos de detección de anomalías buscan identificar patrones o instancias que sean significativamente diferentes de la mayoría de los datos.

1.2. APRENDIZAJE SUPERVISADO

El aprendizaje supervisado es otra categoría dentro del aprendizaje automático o estadístico y, a diferencia del aprendizaje no supervisado, utiliza datos clasificados para la construcción y estimación de los modelos. Es decir, la base de datos con la que se obtiene el modelo contiene observaciones ya clasificadas debido a que el objetivo final es obtener una buena aproximación a

$f(x)$ para todo x y que dicha función pueda generalizar a nuevos datos prediciendo la respuesta correcta. Además, los algoritmos de aprendizaje supervisado tienen la propiedad de poder modificar la relación de *inputs/outputs* $\hat{f}(x_i)$ en respuesta a la diferencia $y - \hat{f}(x_i)$, cuyo proceso es conocido como aprendizaje mediante ejemplos.

En la estimación de los modelos de aprendizaje supervisado se busca minimizar la función de pérdida empírica, la cual es definida como la diferencia entre las predicciones del modelo y los valores reales de los datos de entrenamiento, por lo que mide cuan mal se clasifican los datos en promedio. En el caso cuantitativo, la función de pérdida empírica se expresa generalmente como el error cuadrático medio, que mide la media de los errores al cuadrado definidos como la diferencia entre los valores predichos y los valores reales. En el caso cualitativo, la función de pérdida empírica mide la proporción de observaciones correctamente clasificadas con relación al total.

Al estar el foco puesto en la predicción fuera de la muestra, en el universo de algoritmos de aprendizaje supervisado existen técnicas orientadas a evitar un sobreajuste o subajuste del modelo a los datos de entrenamiento y de esa manera obtener una mayor generalización y mejores resultados predictivos. El desempeño predictivo de los modelos derivados de algoritmos de aprendizaje supervisado es cuantificable mediante diversas métricas aplicadas sobre la muestra de prueba.

1.3. ESTIMACION DE MODELOS DE APRENDIZAJE SUPERVISADO

Las muestras de datos utilizados para la estimación de los modelos de aprendizaje supervisado están compuestas por distintas características asociadas cada una a una variable que las clasifica. La misma puede ser del tipo cuantitativo o cualitativo y de ello dependerá la clase de algoritmo a utilizar y las métricas para evaluar su rendimiento. En general, la muestra de datos completa se divide aleatoriamente en 3 submuestras:

- Muestra de entrenamiento: con esta muestra se estiman los parámetros pertinentes de $f(x)$ y así se obtiene $\hat{f}(x_i)$ para cada algoritmo.
- Muestra de validación: Se utiliza para optimizar los hiperparámetros de cada algoritmo optimizando la métrica de rendimiento escogida.

- Muestra de prueba: con esta muestra se comparan los algoritmos eligiendo aquel que demuestre mejor capacidad predictiva mediante el cálculo de distintas métricas de rendimiento.

La muestra de validación cumple la función de contener datos provenientes de una submuestra aleatoria para la optimización de los hiperparámetros de cada algoritmo. Los hiperparámetros, a diferencia de los parámetros estimados por cada algoritmo, no se estiman directamente de los datos, sino que deben ser seleccionados antes del momento de estimación ya que afectan la capacidad del modelo para aprender de los datos. La selección adecuada del valor de estos hiperparámetros es importante para obtener un modelo preciso y generalizable, por lo que se requiere explorar múltiples combinaciones de valores como la búsqueda en cuadrícula o la optimización bayesiana.

En general, la optimización de los hiperparámetros se lleva a cabo mediante la búsqueda en cuadrícula junto con el uso de la técnica de validación cruzada. Esta técnica consiste en dividir los datos en k grupos donde se utiliza lo utiliza al grupo k como conjunto de prueba y al restante $k-1$ como conjunto de entrenamiento, repitiendo el proceso k veces de modo que cada grupo se utilice una vez como conjunto de prueba para cada una de las combinaciones posibles de hiperparámetros. De esta manera, se obtienen $n \times m$ posibles combinaciones, las cuales son estimadas k veces cada una por los distintos grupos de entrenamiento. Los hiperparámetros óptimos son aquellos para los cuales el modelo estimado maximiza la métrica de desempeño establecida.

Finalmente, para la elección del algoritmo de aprendizaje supervisado que mejor desempeño predictivo presente para el conjunto de datos, se obtienen diversas métricas de rendimiento a partir de la muestra de prueba. Esta aplicación simula tratar con datos de la realidad obtenidos fuera de la muestra, y permite observar el desempeño de cada uno de los algoritmos a la vez que decidir cual actúa mejor.

1.4. METRICAS DE RENDIMIENTO

El presente trabajo está enfocado en la aplicación de algoritmos de clasificación al problema planteado, debido a la naturaleza cualitativa de la variable explicativa. Por lo tanto, esta sección

estará enfocada en la descripción de las métricas de rendimiento exclusivas de algoritmos de clasificación más utilizadas por la literatura actual.

1.4.1. MATRIZ DE CONFUSIÓN

La matriz de confusión es utilizada para resumir el rendimiento del algoritmo de clasificación. Dado que el simple indicador de la proporción de casos correctamente clasificados sobre el total puede ser engañosa, la matriz de confusión sirve para evaluar las ventajas y desventajas del modelo estimado. En general, esta matriz tiene dos dimensiones: la dimensión horizontal muestra las clases predichas por el modelo, mientras que la dimensión vertical muestra las clases verdaderas.

Tabla 1: Matriz de confusión

		Observación	
		Positivos (1)	**Negativos (0)**
Predicción	**Positivos (1)**	**Verdaderos Positivos (TP)**	**Falsos Positivos (FP)**
	Negativos (0)	**Falsos Negativos (FN)**	**Verdaderos Negativos (TN)**

La matriz se encuentra dividida en cuatro categorías: sobre la diagonal principal se encuentran aquellas observaciones de la muestra de prueba que fueron correctamente clasificadas por el modelo, mientras que las restantes dos categorías se encuentran aquellas que no lograron ser correctamente clasificadas. Los falsos positivos (FP) hacen alusión al error de tipo I, mientras que los falsos negativos (FN) al error de tipo II.

De los cuatro elementos que componen la matriz de confusión se derivan una serie de métricas que permiten evaluar el desempeño predictivo del modelo desde distintos puntos de vista. Lo interesante de analizar dichas métricas es que permiten poner el foco de atención en diferentes aspectos de la capacidad predictiva del modelo. En este trabajo se utilizan los ratios de exactitud, especificidad, sensibilidad, precisión, F1-score y kappa.

- **A) Exactitud (Accuracy)**

La exactitud es el ratio más comúnmente utilizado para evaluar el rendimiento de un modelo de aprendizaje automático. Se define como la proporción de observaciones de prueba que el modelo clasifica correctamente.

$$Exactitud = \frac{TP + TN}{TP + FP + FN + TN}.$$

Donde TP y TN representan la cantidad de verdaderos positivos y negativos respectivamente, mientras que FP y FN la cantidad de falsos positivos y negativos.

- **B) Sensibilidad (Tasa de Positivos Reales)**

El ratio de sensibilidad, también conocido como tasa de positivos reales, mide la proporción de verdaderos positivos que son clasificados correctamente. En otras palabras, mide la capacidad del modelo para detectar todos los ejemplos positivos en el conjunto de datos.

$$Sensibilidad = \frac{TP}{TP + FN}.$$

- **C) Precisión**

La precisión mide la proporción de casos positivos predichos que fueron correctamente clasificados como tal. En otras palabras, mide la eficacia del modelo en la clasificación de los casos positivos

$$Precision = \frac{TP}{TP + FP}.$$

- **D) F1 Score**

El F1 score es una métrica de evaluación que combina la precisión y la sensibilidad. Esta métrica es relevante cuando se busca encontrar un equilibrio entre ambas métricas. Se calcula a partir de la media armónica entre la precisión y la sensibilidad.

$$F1\ score = 2 \times \frac{Precision \times Sensibilidad}{Precision + Sensibilidad}.$$

- **E) Kappa**

El coeficiente de Kappa muestra cuanto mejor clasifica un algoritmo en comparación con otro algoritmo que lleve a cabo predicciones aleatorias siguiendo a la frecuencia de cada clase. El coeficiente se calcula como

$$Kappa = \frac{p_0 - p_e}{1 - p_e},$$

donde p_0 es la exactitud del modelo y p_e es la exactitud esperada, es decir, la exactitud que se espera que alcance cualquier clasificador aleatorio basado en la matriz de confusión. El valor de p_0 se calcula como:

$$p_0 = \frac{TP + TN}{TP + FN + FP + TN}.$$

Para calcular p_e son necesarios otros dos ratios: p_1 y p_2. El ratio p_1 representa la probabilidad de que una observación de la muestra sea realmente positiva. A partir de la matriz de confusión se puede calcular p_1 como

$$p_1 = \frac{TP + FN}{TP + FN + FP + TN}.$$

por lo que $(1 - p_1)$ es la probabilidad de que una observación de la muestra sea realmente negativa, mientras que p_2 representa la probabilidad de que el modelo prediga a una observación como positiva. De la matriz de confusión p_2 se puede calcular como

$$p_2 = \frac{TP + FP}{TP + FN + FP + TN},$$

por lo que $(1 - p_2)$ es la probabilidad de que el modelo prediga a una observación como negativa.

Finalmente, p_e se calcula como

$$p_e = p_1 \times p_2 + (1 - p_1) \times (1 - p_2)$$

El valor del coeficiente se encuentra en el rango $[-1,1]$. Si el valor que toma p_e es menor o igual a cero, significa que el clasificador que se esta evaluando no es útil. Para valores mayor a cero, Landis *et al*. (1977) propuso distintas calificaciones en base a los valores: para valores entre 0-0.2 como leve, 0.21-0.4 razonable, 0.41-0.6 moderado, 0.61-0.8 considerable, y 0.81-1 casi perfecto.

1.4.2. CURVA ROC Y AREA BAJO LA CURVA

La curva ROC (*Receiver Operating Characteristic*) es una métrica de evaluación para problemas de clasificación binaria y consiste en una representación gráfica de la tasa de verdaderos positivos (TPR o sensibilidad) frente a la tasa de falsos positivos (FPR), la cual se construye como

$$FPR = \frac{FP}{FP + TN}.$$

y se interpreta como la cantidad de observaciones realmente negativas que el algoritmo clasificó erróneamente.

Cada punto de la curva representa un par de valores de TPR y FPR correspondiente a un umbral de decisión particular. Luego, la curva permite observar gráficamente la capacidad global del modelo para distinguir entre clases.

Gráfico 1: Curva ROC y área bajo la curva ROC

Fuente: Elaboración propia

Como se observa en el gráfico, una curva ROC que se ubique hacia el extremo superior izquierdo corresponde a un clasificador perfecto, es decir, clasifica correctamente todos los datos. Por otro lado, si la curva se ubica sobre la diagonal corresponde a un clasificador aleatorio.

La curva ROC sirve para comparar entre algoritmos, sin embargo, cuando las curvas de dos algoritmos diferentes son muy cercanas o se cruzan el análisis gráfico pierde su utilidad, ya que no es posible determinar cuál es mejor. Es por esto por lo que en general se utiliza el área bajo la curva ROC (AUC-ROC), la cual puede tomar valores dentro del rango [0.5-1], siendo 0,5 el caso del clasificador aleatorio y 1 el caso del clasificador perfecto.

2. ALGORITMOS DE CLASIFICACION

2.1. REGRESION LOGISTICA

La regresión logística es un algoritmo de aprendizaje supervisado utilizado para la clasificación de datos binarios. Su objetivo es predecir la probabilidad de que una determinada muestra pertenezca a una de las dos clases posibles. Para ello modela la relación entre la variable explicada y la explicativa a través de la función logística (o función sigmoide).

Partiendo del modelo clásico de regresión lineal es posible modelar la relación entre la variable dependiente binaria y y las variables regresoras x de manera tal de estimar la probabilidad de observar $y = 1$ dado x. Por lo tanto:

$$p(x) = \Pr(y = 1|x) = x\beta \,, (2.1)$$

donde x es un vector transpuesto de regresores con k componentes y cuya primera componente es igual a 1, mientras que β es un vector de k componentes. Sin embargo, $x\beta$ resulta inconveniente debido a que es posible obtener estimaciones de probabilidad negativa o mayor a 1. Para ello, se aplica la función logística al modelo lineal tal que

$$p(x) = \frac{e^{x\beta}}{1 + e^{x\beta}} \,, (2.2)$$

cuyas probabilidades se encuentran entre 0 y 1. Trabajando algebraicamente (2) y aplicando logaritmos a ambos lados de la igualdad se obtiene la siguiente expresión:

$$\log\left(\frac{p(x)}{1 + p(x)}\right) = x\beta \,. (2.3)$$

Si tenemos una muestra aleatoria de tamaño N, $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, la función predictora para la i-ésima observación (y_i, x_i) viene dada por:

$$\widehat{y_i} = \hat{f}(x_i) = \begin{cases} 1 \;\; si \;\; \dfrac{e^{x_i\hat{\beta}}}{1 + e^{x_i\hat{\beta}}} > h \\ 0 \;\; si \;\; \dfrac{e^{x_i\hat{\beta}}}{1 + e^{x_i\hat{\beta}}} \leq h \end{cases},$$

siendo h el umbral de clasificación definido y $\hat{\beta}$ el estimador de β.

Las componentes del vector β se estiman mediante la maximización de la función de verosimilitud. Teniendo en cuenta que y_i es una variable aleatoria cuya función densidad es

$f(y_i) = p(x_i)^{y_i}\big(1 - p(x_i)\big)^{1-y_i}$ para todo i y que y_i es independiente de y_j para todo $i \neq j$, la función de verosimilitud a maximizar dada las N observaciones resulta:

$$\ell(\beta) = \sum_{i=1}^{N} y_i\,(x_i\beta) - \sum_{i=1}^{N} \log\big(1 + e^{x_i\beta}\big)\,. \quad (2.4)$$

Finalmente, en general para obtener los componentes $\widehat{\beta_0}, \widehat{\beta_1}, \ldots, \widehat{\beta_k}$ del vector $\hat{\beta}$ se igualan las derivadas de $\ell(\beta)$ a cero obteniendo $k + 1$ ecuaciones no lineales en el vector β. Para resolver las ecuaciones correspondientes a las condiciones de primer orden se utiliza el algoritmo *Newton-Raphson*.

Sin embargo, en el contexto del aprendizaje automático y considerando que el objetivo principal de este es la construcción de modelos cuya predicción fuera de la muestra sea óptima, la capacidad de generalización de los modelos es un elemento importante al momento de la estimación de los coeficientes. En este sentido, los coeficientes estimados por el método de máxima verosimilitud pueden no presentar resultados satisfactorios debido a un sobreajuste a los datos de entrenamiento.

Para el presente trabajo los coeficientes del modelo construido a partir del algoritmo de Regresión Logística serán estimados utilizando el método de máxima verosimilitud adicionando un término de penalización que busca controlar la complejidad del modelo acercando o llevando algunos coeficientes a cero. Existen 3 posibles penalizaciones a utilizar: L1, L2 y Elasticnet.

- L1: Los coeficientes estimados surgen de resolver el siguiente problema:

$$\max_{\beta}\left\{\sum_{i=1}^{N} y_i(x_i\beta) - \log\big(1 + e^{x_i\beta}\big) - \lambda\|\beta\|_1\right\},$$

representando $\|\beta\|_1$ el cuadrado de la norma 1 del vector β y λ un número real fijo.

En este caso para $\lambda > 0$ se obtienen soluciones de esquina en donde algunos coeficientes del modelo llegan a cero. Si $\lambda = 0$ entonces no hay penalización hacia los coeficientes estimados y estos son los mismos que en (2.4).

- L2: Los coeficientes estimados surgen de resolver:

$$\max_{\beta}\left\{\sum_{i=1}^{N} y_i(x_i\beta) - \log\left(1 + e^{x_i\beta}\right) - \lambda\|\beta\|_2^2\right\},$$

representando $\|\beta\|_2^2$ el cuadrado de la norma 2 del vector β. A diferencia del caso anterior, las soluciones a este problema son siempre interiores

- Elasticnet: La función a minimizar incluye ambas penalizaciones L1 y L2 juntas.

$$\max_{\beta}\left\{\sum_{i=1}^{N} y_i(x_i\beta) - \log\left(1 + e^{x_i\beta}\right) - (\lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2)\right\},$$

siendo λ_1 y λ_2 dos números reales positivos.

2.2. SUPPORT VECTOR MACHINE

El algoritmo *Support Vector Machine (SVM)* es un algoritmo de aprendizaje supervisado que se utiliza para la clasificación y regresión de datos binarios. La idea principal detrás del algoritmo SVM es encontrar un hiperplano que maximice la distancia entre las clases en el espacio de características. En otras palabras, el algoritmo SVM busca una separación clara entre las clases para hacer predicciones precisas. El hiperplano que separa las clases[2] entre $y_i = 1$ o $y_i = -1$ se define como

$$x\delta = 0, (2.5)$$

[2] Para este algoritmo las observaciones clasificadas como 0 en la base de datos original se reclasifican como -1 para su funcionamiento

donde δ es un vector normal al hiperplano, y (2.5) representa una ecuación general del mismo. A priori y de ser posible, la regla de clasificación para la observación i dado su vector de características x_i es:

$$\widehat{y_i} = \begin{cases} 1 & si\ x_i\delta \geq 1 \\ -1 & si\ x_i\delta < 1 \end{cases},$$

y de manera general se debe cumplir que

$$y_i(x_i\delta) - 1 \geq 0. \quad (2.6)$$

Por otro lado, la distancia entre el hiperplano y los puntos de datos más cercanos se conoce como margen. El algoritmo SVM busca maximizar este margen para encontrar la mejor separación entre las clases. Para cualquier observación (x_i, y_i) que se encuentre dentro de este margen se verifica que:

$$y_i(x_i\beta) - 1 = 0. \quad (7)$$

Sin embargo, existe la posibilidad de que las clases no sean separables por ningún hiperplano, debido a que los datos presentan algún tipo de relación no lineal. En este caso, la solución viene dada por la transformación de los datos contenidos en el vector de características x de forma tal que dicha transformación permita obtener una frontera de clasificación lineal sobre las características transformadas. Esta transformación se lleva a cabo mediante la utilización de una función k denominada kernel.

Teniendo en cuenta que $\langle\cdot,\cdot\rangle$ representa el producto interior entre dos vectores, las funciones kernel consideradas en este trabajo son:

- Lineal:
  $k(x_i, x_j) = \langle x_i, x_i\rangle.$

- Polinomico:
  $k(x_i, x_j) = (1 + \langle x_i, x_i\rangle)^d$ donde d es la potencia a la que se eleva el producto punto.

- Radial:

$$k(x_i, x_j) = \exp\left(-\gamma \sum_{j=1}^{p} \|x_i - x_j\|_2^2\right) \text{ donde el parámetro } \gamma \text{ controla el ancho del kernel.}$$

Luego, para obtener la función predictora se resuelve el siguiente problema de optimización utilizando la muestra de entrenamiento:

$$\min_{\{\alpha,\xi\}} \frac{1}{2}\|\alpha\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{sujeto a: } y_i\left(\alpha^T K(x_i)\right) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0,$$

donde $K(x_i)$ es un vector que $n(n-1)/2 + 1$ componentes, cuya $j-$ésima componente es $k(x_i, x_j)$ y su última componente toma valor 1, α es un vector del mismo tamaño de $K(x_i)$ cuyas componentes y ξ_i representa una variable de holgura que tomará un valor positivo cuando la observación i se clasifique incorrectamente. Luego, C es un hiperparámetro de regularización y se obtiene mediante validación cruzada junto con los hiperparámetros de cada kernel (d y γ para el kernel polinómico y radial respectivamente).

Finalmente, la función predictora viene dada por

$$\widehat{y_i} = \hat{f}(x_i) = I_{\{x\,|\alpha^T K(x_i) \geq 0\}}(x_i) - I_{\{x\,|\alpha^T K(x_i) < 0\}}(x_i)\,,$$

donde I hace referencia a la función indicadora que clasifica los datos entre valores en $\{-1,1\}$ a partir del signo que resulte de $\alpha^T K(x_i)$ para cada vector de entrenamiento x_i.

2.3. K NEAREST NEIGHBOURS

El algoritmo clasificador de *K-Nearest-Neighbours* se basa en la identificación de las K observaciones pertenecientes a la muestra de entrenamiento más cercanas a una observación

x_0 perteneciente a la muestra de prueba. Luego, se determina la clase más frecuente entre las K observaciones obtenidas y se le asigna dicha clase a la observación x_0.

En el caso de clasificación binaria, para los K vecinos más cercanos en la muestra de entrenamiento, el predictor y_0 estima la probabilidad condicional para la clase 1 como la fracción de observaciones en K para la cual $y_j = 1$

$$\widehat{y_0} = \begin{cases} 1 \; si \; \widehat{\Pr}(y_0 = 1 | x = x_0) = \dfrac{1}{K} \displaystyle\sum_{j \epsilon K} I\left(y_j = 1\right) > 0.5 \\ 0 \; si \; \widehat{\Pr}(y_0 = 1 | x = x_0) = \dfrac{1}{K} \displaystyle\sum_{j \epsilon K} I\left(y_j = 1\right) < 0.5 \end{cases}.$$

En general, la clasificación de un vector de características x_i depende del predictor y_i que viene dado por la siguiente expresión:

$$\widehat{y_i} = \begin{cases} 1 \; si \; \dfrac{1}{K} \displaystyle\sum_{j \epsilon K} y_j > 0.5 \\ 0 \; si \; \dfrac{1}{K} \displaystyle\sum_{j \epsilon K} y_j < 0.5 \end{cases},$$

donde j es vecino de i si la distancia entre los vectores de características x_i y x_j está entre las menores de toda la muestra de entrenamiento.

Para la medición de las distancias se prueban las 3 más conocidas: Euclidia, Manhattan y Distancia del supremo. No obstante, es posible intentar buscar un valor óptimo para la siguiente función de distancia

$$d(x_i, x_j) = \left(\sum_{t=1}^{p} \left| x_i^t - x_j^t \right|^q \right)^{\frac{1}{q}}, \quad q \geq 1$$

donde x_i^t hace referencia a la t-ésima característica observada del sujeto i. Esta distancia se conoce como distancia de Minkowski y las distancias antes mencionadas son casos particulares de esta última donde se considera $q = 2$, $q = 1$ y $q \to \infty$.

2.4. RANDOM FOREST

El algoritmo de *Random Forest* parte de la construcción de modelos de árboles de decisión. Estos modelos consisten en la división binaria de forma recursiva de la muestra de entrenamiento en base a una característica determinada, obteniendo distintos nodos internos y un total de l nodos terminales u hojas. En cada una de las hojas del árbol de decisión se obtienen distintas proporciones entre las clases correspondientes a la muestra de entrenamiento, por lo que la función de predicción clasificará a la observación i que caiga en la hoja k la clase cuya proporción sea mayor.

Durante la construcción del árbol de decisión se utiliza algún criterio para dividir el nodo k en dos nodos κ_L y κ_R en base a una característica observable x^j y un margen t de forma tal que

$$\kappa_L(t,j) = \{i \mid x_i^j \leq t\}$$
$$\kappa_R(t,j) = \{i \mid x_i^j > t\}$$

El criterio utilizado para la división mide la heterogeneidad o impureza de κ_L y κ_R a través de la proporción de individuos que pertenezcan a la clase 1 y se encuentren en el nodo k denominada como p_κ. Luego, definimos

$$D_\kappa^E = -|\kappa|(p_\kappa \log p_\kappa + (1-p_\kappa)\log(1-p_\kappa)),$$
$$D_\kappa^G = -|\kappa| 2 p_\kappa (1-p_\kappa),$$

donde D_κ^E es la heterogeneidad medida a través de la Entropía del nodo κ y D_κ^G es la heterogeneidad medida a través de la concentración de Gini del nodo κ. De esta forma, las variables j y t se eligen de forma tal que minimicen

$$D_{\kappa_L} + D_{\kappa_R},$$

donde D_{κ_L} y D_{κ_R} son las medidas de heterogeneidad del nodo izquierdo y el nodo derecho en la partición a realizar.

Este proceso se lleva a cabo de manera recursiva de nodo a nodo hasta obtener un nodo completamente homogéneo o que se cumpla con algún criterio de interés como cantidad mínima de observaciones por nodo, o profundidad del árbol.

Uno de los problemas del modelo de árbol de decisión es que sufren de varianza alta, lo cual implica un problema de robustez en las predicciones. Es por esto por lo que se implementa el algoritmo de *Random Forest*, el cual logra reducir la varianza y obtener predicciones más robustas mediante la estimación de un numero finito de árboles de decisión de la siguiente forma:

- Se obtiene una submuestra por *bootstraping* con reemplazo de la muestra de entrenamiento.
- Se eligen aleatoriamente m variables explicativas del conjunto de p. En general, se selecciona $m = \sqrt{p}$.
- Se construye un árbol óptimo con la submuestra y las m variables explicativas seleccionadas. La construcción se lleva a cabo por el proceso descripto anteriormente.
- El proceso se repite una cantidad finita de veces.

El predictor obtenido por el algoritmo *Random Forest* es el resultado de la "votación", es decir, que si en la mayoría de los árboles el predictor toma valor 1 se le asigna valor 1.

2.5. GRADIENT BOOSTING MODEL

Los modelos *boosting* presentan un enfoque general que se puede aplicar a distintos métodos de aprendizaje estadístico para clasificación. Este estudio se enfoca en el contexto aplicado a arboles de decisión.

En este enfoque los árboles son creados de manera secuencial, es decir, cada árbol se construye utilizando información de árboles previos y, a diferencia del algoritmo *Random Forest*, cada árbol se construye a partir de la muestra de entrenamiento original. Se explica a continuación una forma sencilla de implementación[3].

[3] Para un desarrollo completo dirigirse a Hastie et. al (2009).

- Se estima el mejor árbol posible para la totalidad de datos de la muestra de entrenamiento. Del mismo se obtiene el predictor $\widehat{f_0}$ y las observaciones mal clasificadas.
- Se estima el mejor árbol posible sobre las observaciones mal clasificadas y a su predictor se denomina como $\widehat{f'}_1$.
- Se actualiza el predictor para la totalidad de la muestra de la siguiente forma

$$\widehat{f_1} = \frac{1}{2}\widehat{f_0} + \frac{1}{2}\widehat{f'_1}$$

  y se computan las nuevas observaciones mal clasificadas
- Se repite el proceso una cantidad m de veces

La idea detrás del funcionamiento de este algoritmo es que en cada iteración el predictor se actualiza y genera predicciones más certeras. De todos los algoritmos de este estilo se utiliza *XGBoost*.

3. PLATAFORMAS P2P

Los prestamos persona a persona o *peer to peer lending* son redes financieras que permiten al inversor otorgar como préstamo a consumidores/prestatarios de fondos a través de una plataforma intermediaria. Sin embargo, la definición exacta de que son las plataformas P2P ha sido discutida por distintos autores. Moenninghoff y Wieandt (2012) definen a estas plataformas como "el proceso de provisión directa de préstamo por parte de los prestadores a través de internet", mientras que Brent King *et al.* (2010) hace foco en la diferencia respecto al sistema tradicional afirmando que estas plataformas "no toman realmente depósitos, sólo facilitan el acuerdo para llegar a un arreglo contractual entre dos personas y cobran pequeñas comisiones como intermediarias".

El crecimiento de estas plataformas se da en el contexto de la crisis financiera de 2007/2008 y la consecuente pérdida de confianza en las grandes y tradicionales instituciones financieras. Las precursoras fueron Zopa en Inglaterra, fundada en 2005, y Prosper y Lending Club en Estados

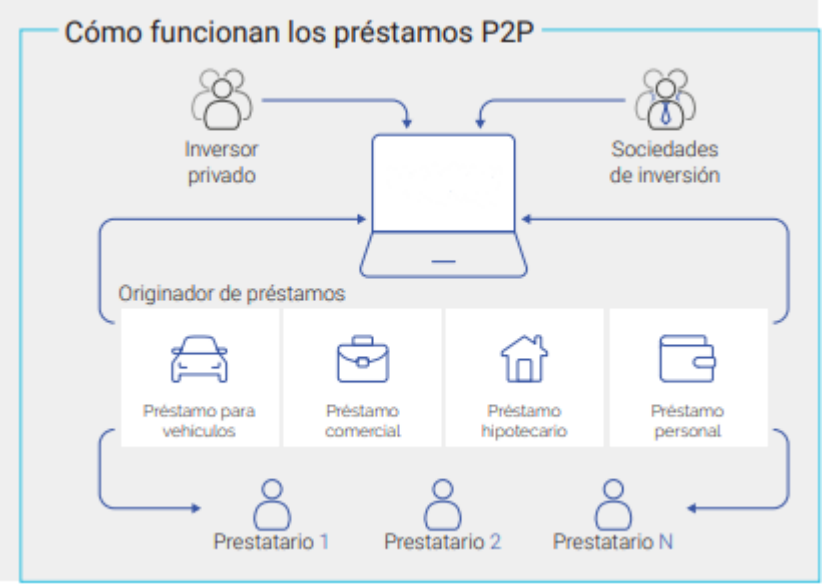
Unidos, fundadas en 2006 y 2007 respectivamente. A partir de 2008 comienza la proliferación de compañías y el fuerte crecimiento en los montos invertidos.

El modo de funcionamiento de este tipo de negocios funciona de la siguiente manera: prestamistas o inversores se registran en alguna de las plataformas disponibles y realizan la transferencia del monto que quieren prestar. Por otro lado, aquellos que quieran solicitar un préstamo también se registran y realizan una solicitud de crédito detallando el motivo de la solicitud y facilitando datos financieros que permitan asignar un puntaje crediticio. Cada una de las plataformas tiene su propio sistema de evaluación de riesgo crediticio, y de acuerdo con los datos de cada solicitud asigna una categoría de riesgo siendo A la categoría más alta, o directamente se excluye por no cumplir con los requisitos mínimos. Una vez aprobada la solicitud de crédito, esta se publica en el *Marketplace* de cada plataforma donde se busca la contraparte que otorgue los fondos. En general, las plataformas diversifican la inversión en varios préstamos diferentes para atomizar el riesgo.

Una de las características esenciales de los prestamos generados en estas plataformas es que la tasa de interés pactada es propia de cada acuerdo, por lo que quienes solicitan préstamos son generalmente individuos que buscan refinanciar deuda a tasas de interés razonables, o pequeñas y medianas empresas que tienen problemas para conseguir prestamos en instituciones financieras tradicionales. Los inversores, quienes proveen los fondos de los préstamos y reciben retornos sobre el capital invertido, pueden ofrecer tasas atractivas ya que el único costo que deben afrontar hacia la plataforma es una relativamente baja comisión por el valor generado.

A continuación, en el gráfico 2 se puede observar el proceso de aplicación y colocación de un crédito en estas plataformas de forma resumida.

Gráfico 2: Funcionamiento de las plataformas P2P



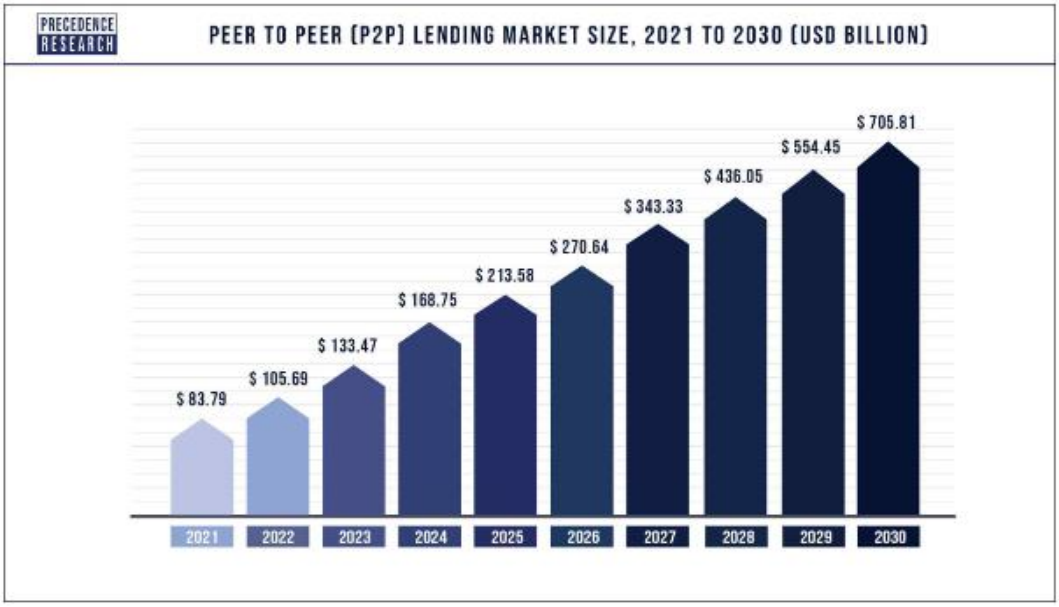
Fuente: Grupeer

En relación con el riesgo que acarrean estas plataformas, Moenninghoff y Wieandt (2012) indican que las plataformas P2P llevan asociados varios riesgos financieros entre los que se encuentran el riesgo de crédito, tasa de interés, mercado y liquidez. Además, las características del funcionamiento de estas plataformas requieren que los participantes asuman la totalidad del riesgo, generando incentivos inadecuados debido a que el mismo no es internalizado por las plataformas al momento de la maximización de beneficios. Por esto, mitigar los riesgos asociados es una tarea clave para el sano crecimiento de estas plataformas, y es asi que se llevan a cabo prácticas como la diversificación de las carteras y mejoras constantes de sus modelos de scoring crediticio derivados de algoritmos de aprendizaje automático.

3.1. ACTUALIDAD Y PROYECCIONES A FUTURO

Según Precedence Research (2022), durante el 2021 el tamaño de mercado global de las plataformas P2P alcanzaba los 83 billones de dólares y pronostica un crecimiento hacia 2030 en el que alcanzaría los 705 billones de dólares.

Gráfico 3: Market size de las plataformas P2P

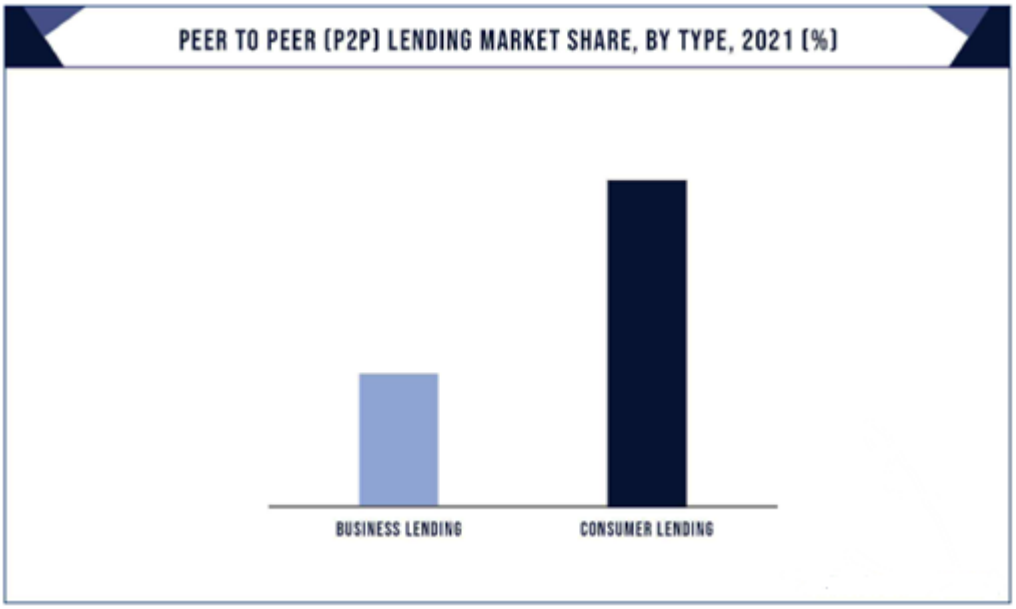


Fuente: Precedence Research (2022)

Entre los factores que enumera como claves del crecimiento pronosticado se encuentran las menores comisiones en relación con el sistema tradicional, la rápida aprobación de los créditos, la mayor digitalización del sector bancario y por consiguiente la mayor facilidad de transferencias, la mayor demanda de fondos por parte de las pequeñas y medianas empresas, el avance tecnológico en relación con el internet de las cosas y *blockchain*, entre otras. Por otro lado, factores como las mayores regulaciones gubernamentales a las llamadas *Fintech* y la falta de conocimiento de estas plataformas se mencionan como limitantes de su crecimiento.

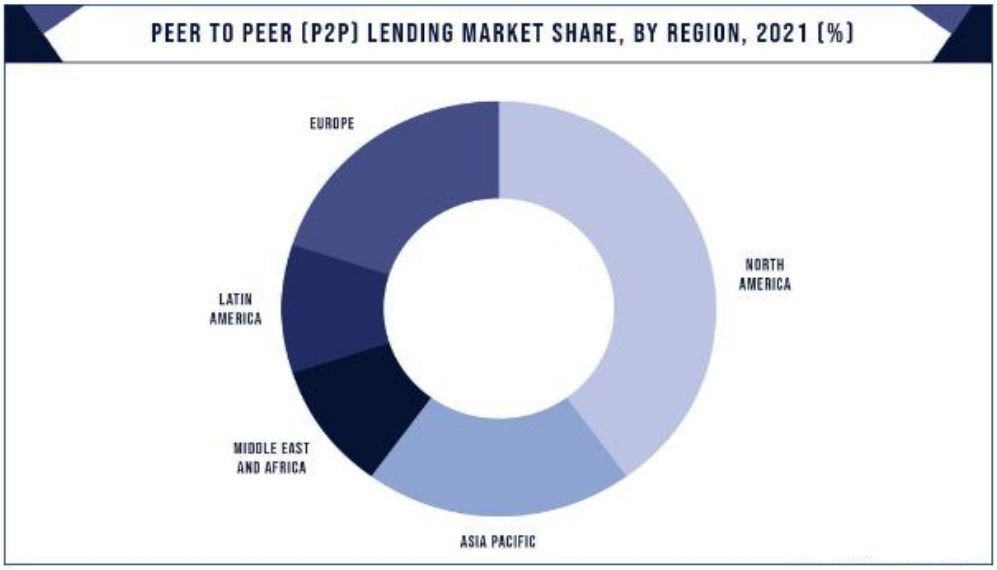
Hacia 2021, según el mismo reporte, el segmento de consumidores del tipo individuos era dominante. Sin embargo, el segmento de préstamos a empresas fue el de mayor crecimiento hacia ese momento traccionado por el aumento en la demanda por parte de pequeñas y medianas empresas y *start ups*.

Gráfico 4: Participación por cliente de las plataformas P2P



Fuente: Precedence Research (2022)

Gráfico 5: Participación por pais de las plataformas P2P



Fuente: Precedence Research (2022)

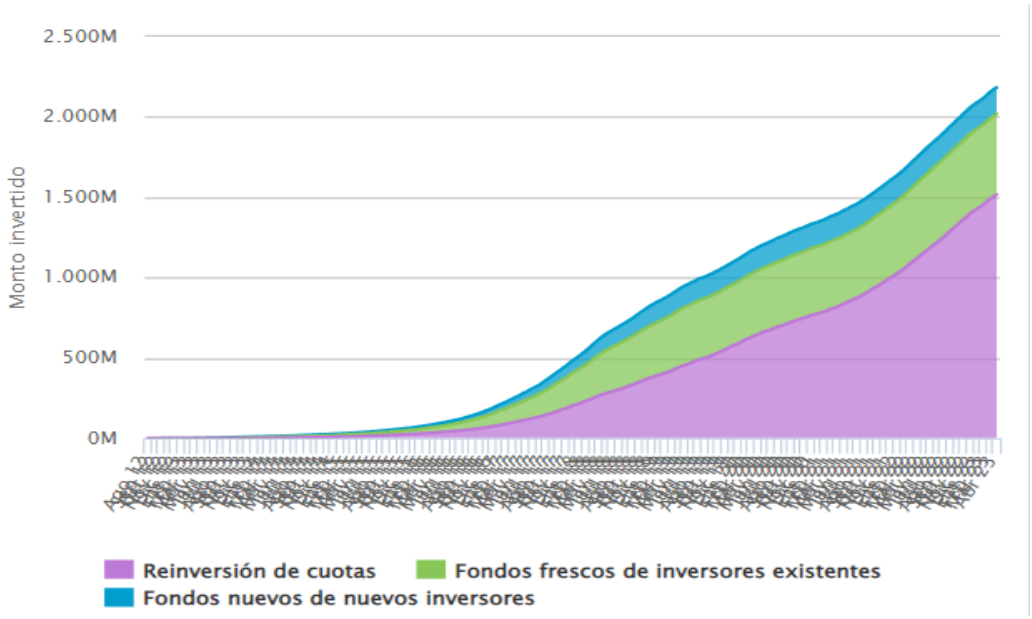
En el análisis por región, America del Norte fue la que dominó el mercado global de préstamos a través de plataformas P2P. Durante el periodo proyectado, se espera que la región de Asia Pacífico sea la que mayor tasa de crecimiento presente debido a los esfuerzos por parte de los

gobiernos tales como China e India en la promoción de tecnologías sin uso de efectivo. La región de Latinoamérica junto con Medio Oriente y África son las que menos avanzadas están en el uso de este tipo de plataformas de crédito.

3.2. PLATAFORMAS P2P EN ARGENTINA: CASO AFLUENTA

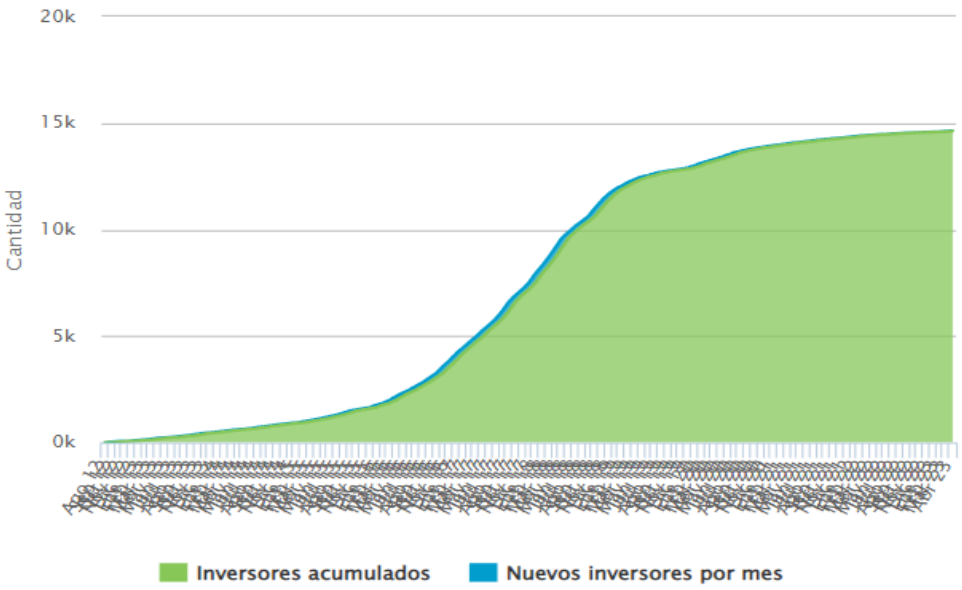
Afluenta es una compañia que forma parte de las denominadas finanzas colaborativas fundada en Argentina en 2011. La compañía comienza sus operaciones como plataforma de préstamos P2P en 2012 y a partir de allí presentó un crecimiento rápido y constante de fondos hasta la actualidad, y una desaceleración en la cantidad de inversores a partir del 2019.

Gráfico 6: Ingreso mensual de fondos a Afluenta



Fuente: Afluenta

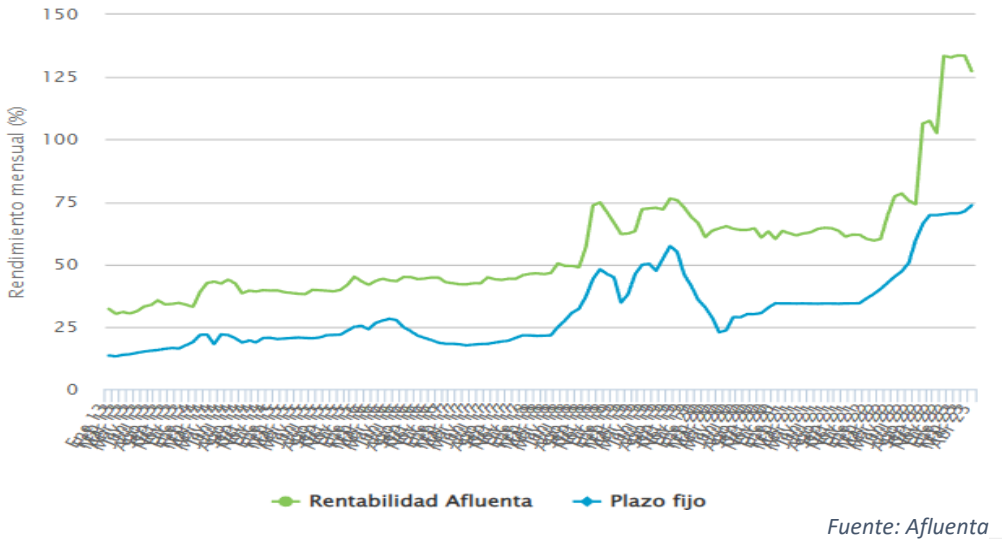
Gráfico 7: Evolución mensual de inversores de Afluenta



Fuente: Afluenta

Afluenta se posiciona como una opción atractiva para los inversores en relación con una inversión convencional en un plazo fijo bancario. En Abril del 2023 un plazo fijo rendía una TNA alrededor del 74%, mientras que el capital invertido en Afluenta rendía en promedio una TNA de 127%. Sin embargo, esta tasa es variable dependiendo del segmento de cliente al cual se le presta pudiendo aumentar hasta 147% para los segmentos más riesgosos. El *spread* de tasas promedio se presenta a continuación

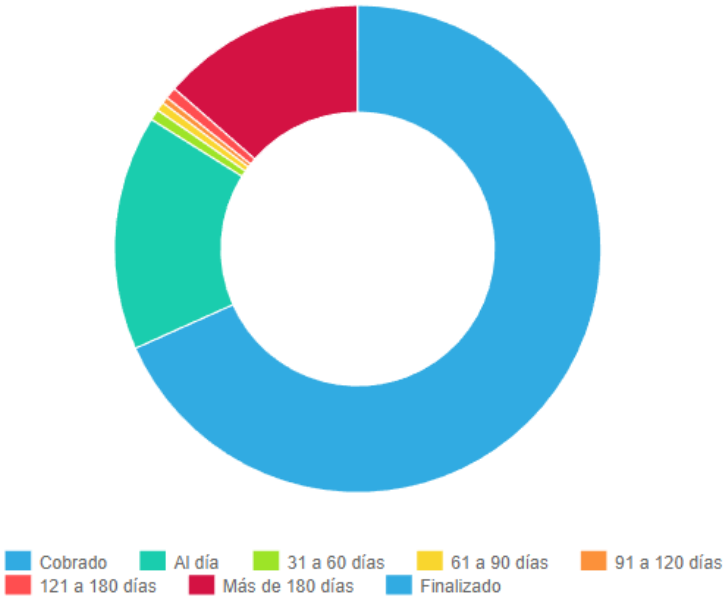
Gráfico 8: Evolución mensual del retorno sobre capital invertido en Afluenta



Fuente: Afluenta

Con respecto al estado de la cobranza de los préstamos en Afluenta, del total de préstamos casi el 70% fue cobrado, alrededor del 15% esta al día y el restante 15% se encuentra en situación de mora, llegando a representar casi el 14% aquellos prestamos con una mora mayor a 180 días. Este valor resulta importante, teniendo en cuenta que la irregularidad del crédito al sector privado de Febrero de 2023 para el caso de empresas publicado por el Banco Central fue de 3,3%.

Gráfico 9: Estado de la cobranza de los préstamos en Afluenta



Fuente: Afluenta

CAPITULO II
ESTIMACION Y COMPARACION DE ALGORITMOS

1. PRESENTACION Y PREPARACION DE LOS DATOS

1.1. PRESENTACION

La base de datos a utilizar en el presente trabajo de investigación es construida por sus autores en Guidici *et al*. (2019) a partir de la información provista por las "European External Credit Assessment Institutions (ECAI)", las cuales son agencias de puntaje crediticio reconocidas en la Union Europea. En este caso, los datos provistos son por parte de aquellas agencias especializadas en el puntaje crediticio de empresas PyME que forman parte de plataformas P2P.

La base de datos utilizada en este estudio está compuesta por ratios construidos a partir de los Estados Financieros[4] oficiales proporcionados por 4514 pequeñas y medianas empresas italianas correspondiente al año 2015. En la misma base de datos se identifica para cada una de las empresas si incurrió en incumplimiento de pago a través de la variable "status", que toma el valor de 1 en caso de default y 0 en caso contrario.

Tabla 2: Resumen estadistico de la base de datos

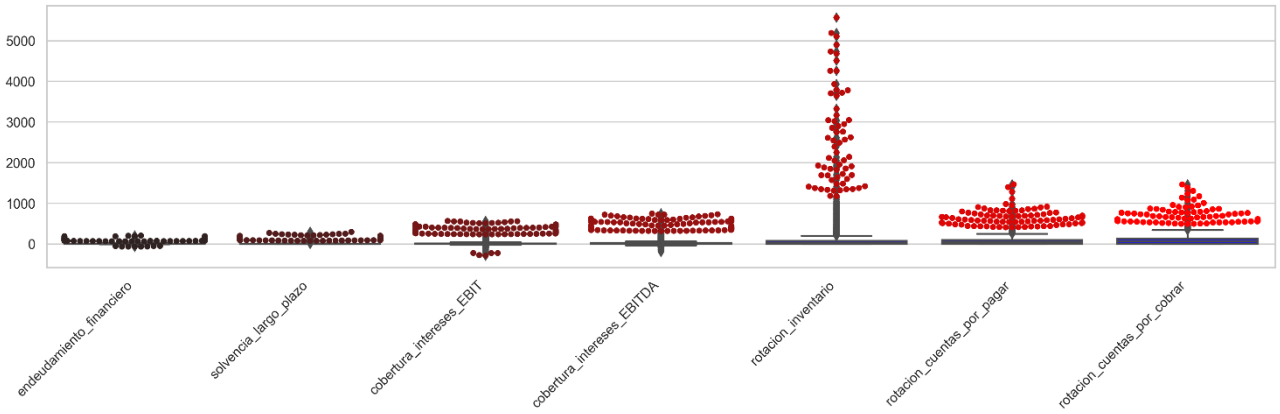
Variable	N	Mean	St. Dev	Min	Max
endeudamiento_financiero	4514	8.89	19.15	-64.43	206.55
endeudamiento_largo_plazo	4514	1.26	3.33	-9.58	33.38
solvencia	4514	1.44	0.76	0.17	8.27
liquidez	4514	1.54	1.2	0.01	13.71
prueba_acida	4514	1.19	1.02	0	10.88
solvencia_largo_plazo	4514	7.93	23.15	0	297.02
cobertura_intereses_EBIT	4514	23.07	70.27	-285.86	566.96
ROA	4514	0.03	0.15	-1.28	0.49
ROE	4514	-0.07	0.79	-8.54	1.08
rotacion_activos_gcia_bruta	4514	1.37	1.07	0.01	8.42
rotacion_activos_ventas	4514	1.34	1.06	0.01	8.42
carga_intereses	4514	0.19	0.5	-3.32	3.95
cobertura_intereses_EBITDA	4514	36.51	92.89	-191.63	747.01
margen_EBITDA_gcia_bruta	4514	0.06	0.2	-2.08	0.94
margen_EBITDA_ventas	4514	0.07	0.22	-2.39	1.28
rotacion_inventario	4514	105.23	355.81	0	5569
rotacion_cuentas_por_pagar	4514	75.93	111.65	0	1467
rotacion_cuentas_por_cobrar	4514	95.73	128.37	0	1465
status	4514	0.11	0.31	0	1

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

[4] Los ratios financieros utilizados son típicos del análisis contable. Para una definición formal de los mismos dirigirse a Dumrauf (2010).

La Tabla 1 proporciona un resumen estadístico de las variables ratios que son utilizadas, de la cual es notable el alto desvío estándar para el caso de ratios como rotación, en sus tres casos y cobertura_intereses_EBITDA, además de la presencia de *outliers* observables mediante las estadísticas de mínimos y máximos en algunas variables que se alejan fuertemente de la media.

Grafico 10: Boxplot con outliers según z-score



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

En el Gráfico 1 se lleva a cabo la identificación de los *outliers* presentes en la base de datos. Los puntos rojos se clasifican como *outliers* en función del resultado del método *z-score* aplicado a cada variable. Este método se utiliza para detectar valores atípicos basándose en la media $(\bar{x})$ y desviación estándar (s) de los datos para la estandarización de los mismos y su comparación. Para cada observación i de cada variable j se calcula su *z-score* mediante la siguiente fórmula

$$z_{ij} = (x_{ij} - \bar{x}_j)/s_j$$

donde s_j es el desvío estándar de la variable j y $\bar{x}_j$ es el promedio de la variable j.

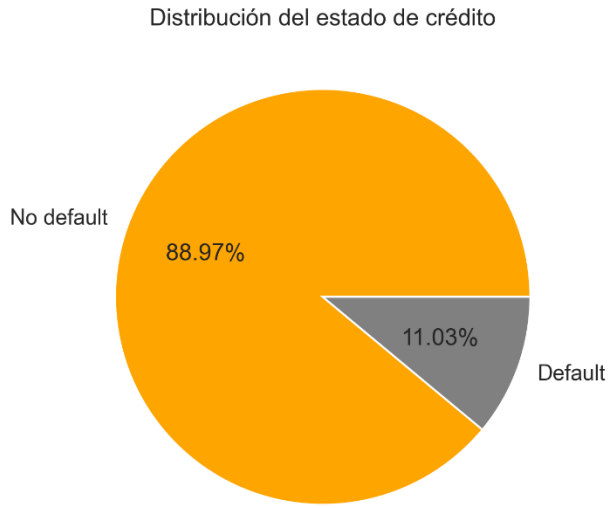
Una vez estandarizada cada variable se puede identificar cuales valores están lejos de la media. Una regla empírica común es considerar "anómalos" aquellos valores para los cuales $|z_{ij}| > 3$.

Como afirma Guidici *et al*. (2019), en la mayor parte de bases de datos basadas en la realidad, especialmente aquellas relacionadas con start-ups y pequeñas y medianas empresas existe una notable presencia de valores inusualmente grandes o pequeños en comparación con la media.

Sin embargo, el tratamiento de estos *outliers* implica fuertes supuestos acerca del tamaño y la distribución de los datos, así como la aleatoriedad de dichos valores atípicos por lo que no es conveniente eliminarlos ni sustituirlos por otros valores.

De la Tabla 1 también es posible notar la naturaleza binaria de la variable *status*, la cual clasifica aquellas empresas que incurrieron en incumplimiento de pago.

Grafico 11: Proporción de empresas en default



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Del gráfico 2 se puede notar el desbalance que presenta la base de datos entre aquellas empresas en situación de *default* y aquellas que no.

Es interesante analizar las diferencias que presentan en sus ratios financieros las empresas en *default* respecto a aquellas que se encuentran en una situación regular de pagos. Para ello se lleva a cabo un test de diferencia de medias a dos colas entre empresas en *default* y no *default* para cada ratio de la base de datos. En la siguiente tabla se encuentran aquellas variables cuyas medias entre clases son significativamente diferentes a partir del p-valor asociado al estadístico *t-student* estimado para cada test. Se considera a las diferencias de medias estadísticamente significativa si el p-valor resulta menor al nivel de significatividad de 0.05. Los gráficos con las distribuciones de cada ratio se encuentran en el Anexo II.

Tabla 3: Resultado test de medias

Variable	Media clase 0	Media clase 1	t-statistic	p-value
endeudamiento_financiero	8.85	9.14	-0.32	0.74
endeudamiento_largo_plazo	1.24	1.38	-0.88	0.37
solvencia	1.48	1.08	11.24	<0.001
liquidez	1.59	1.04	9.83	<0.001
prueba_acida	1.24	0.75	10.11	<0.001
solvencia_largo_plazo	7.99	7.37	0.57	0.57
cobertura_intereses_EBIT	40.17	6.95	7.57	<0.001
ROA	0.04	-0.13	28.24	<0.001
ROE	0.008	-0.69	19.51	<0.001
rotacion_activos_gcia_bruta	1.38	1.3	1.55	0.12
rotacion_activos_ventas	0.21	0.05	1.07	0.28
carga_intereses	0.21	0.05	6.85	<0.001
cobertura_intereses_EBITDA	40.17	6.95	7.57	<0.001
margen_EBITDA_gcia_bruta	0.08	-0.11	23.05	<0.001
margen_EBITDA_ventas	0.09	-0.11	21.59	<0.001
rotacion_inventario	100.61	142.47	-2.47	0.01
rotacion_cuentas_por_pagar	67.34	145.18	-15.03	<0.001
rotacion_cuentas_por_cobrar	91.07	133.31	-6.96	<0.001

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

A partir de la información contenida en la Tabla 2 es posible obtener conclusiones relevantes acerca de las diferencias financieras existentes entre las empresas del tipo 1 (default) y 0 (no default) que forman parte de la base de datos. Las empresas PyMES en *default*:

- Presentan en promedio menores valores para los ratios de solvencia, liquidez, prueba acida, cobertura de intereses por EBIT y EBITDA, ROA, ROE, carga de intereses, margen EBITDA para la ganancia bruta como para las ventas.
- Presentan en promedio mayores valores para los ratios de rotación de inventario, de cuentas por pagar y de cuentas por cobrar.

Además, haciendo foco en los ratios de solvencia y liquidez (principales al momento de llevar a cabo un análisis financiero), resulta que para el caso de las empresas en *default* estos ratios son en promedio un 37% y 57% menor respectivamente.

1.2. PREPARACION

Recordando que el objetivo principal del aprendizaje automático es lograr una buena capacidad predictiva del modelo entrenado y que el problema de investigación del presente trabajo se encuadra dentro del universo del aprendizaje supervisado, es que se deben llevar a cabo ciertos tratamientos a los datos para lograr maximizar el rendimiento predictivo de los modelos estimados (o de igual manera minimizar el error de estos).

De acuerdo con Vinay (2021), la presencia de variables con amplios rangos de variación puede generar problemas en el proceso de estimación del modelo para la mayoría de los algoritmos antes mencionados. Esto se debe a que, al no estandarizar las variables, el algoritmo puede asignar un peso excesivo a aquellas características con mayor variabilidad en los datos, lo que puede afectar negativamente el rendimiento del modelo en términos de precisión y generalización. La estandarización de las variables permite que cada una de ellas tenga la misma escala y distribución, lo que ayuda a garantizar que todas las características sean igualmente importantes durante el proceso de modelado.

Se decide utilizar el método de escalado conocido como *min-max scaler*, el cual consiste en transformar los datos originales de manera que se ubiquen dentro del rango [0,1]. Dicho método de escalado conduce a una reducción de la desviación estándar de los datos, lo que a su vez minimiza el impacto de los valores atípicos. Asimismo, dado que no se modifica la forma de la distribución de los datos, se evita la pérdida de información asociada con otras técnicas de escalado. Para cada observación i de cada variable j, x_{ij}, se aplica la siguiente transformación[5]:

$$x_{ij}^{scaled} = (x_{ij} - x_j^{min}) / (x_j^{max} - x_j^{min})$$

Finalmente, como último paso en la preparación de los datos para el desarrollo de los modelos se divide aleatoriamente la base completa en dos principales sub-bases: la base de entrenamiento – validación con el 70% de los datos y la base de prueba con el restante 30%. Asimismo, la base de entrenamiento – validación se divide en un 70% para la base de entrenamiento

[5] Los gráficos de distribución de los datos escalados se encuentran en el Anexo II.

y un 30% para la base de validación. Tal como indican sus nombres, las bases de datos se utilizarán en distintas etapas del desarrollo de cada uno de los modelos.

2. ESTIMACION DE MODELOS

Para la estimación de modelos a partir de los algoritmos de aprendizaje automático seleccionados, se lleva a cabo la optimización de los hiperparámetros de cada uno de estos. En cada algoritmo se busca maximizar dos métricas:

- Se nombra modelo 1 al que optimiza el *balanced accuracy*. A diferencia del *accuracy* (o exactitud) convencional, esta tiene en cuenta el desequilibrio entre las clases en el conjunto de datos. Matemáticamente es la media aritmética de la sensibilidad o tasa de verdaderos positivos (en adelante TPR o Tasa de Verdaderos Positivos) y la especificidad o tasa de verdaderos negativos (en adelante TNR o Tasa de Negativos Reales) y se expresa como un valor entre 0 y 1.

- El modelo 2 maximiza el área bajo la curva ROC (AUC ROC o área bajo la curva ROC).

La optimización de los hiperparámetros se lleva a cabo mediante la técnica denominada *Grid Search Cross Validation* o la búsqueda en cuadrícula por validación cruzada. En esta técnica, se combinan diferentes valores de hiperparámetros y se evalúa el rendimiento del modelo para cada una de estas combinaciones mediante validación cruzada, el cual consiste en dividir la totalidad de los datos en k partes iguales, utilizando una parte como conjunto de prueba y las restantes k-1 partes como conjunto de entrenamiento. De esta manera, se obtienen k estimaciones diferentes para cada combinación posible de hiperparámetros.

El resultado final de la técnica *Grid Search Cross Validation* es una lista de los mejores valores de hiperparámetros encontrados durante el proceso, es decir, aquellos que maximizan las métricas seleccionadas[6].

[6] A modo de ejemplo, si se define $k = 5$ y hay dos hiperparámetros $A \in (0,1), B \in (0,1)$, dado que A y B no pueden tomar simultáneamente el valor 0 se busca el par (A,B) óptimo en el conjunto

Una vez concluida la etapa de optimización y estimación de los modelos para cada uno de los algoritmos considerados, se evalúa el desempeño predictivo de cada modelo mediante la base de validación. Específicamente, se busca seleccionar aquel modelo que presente el mejor rendimiento para todos los umbrales de clasificación posibles, es decir, el que presente una mayor área bajo la curva ROC. De esta forma, se busca seleccionar aquel modelo que presente la mejor capacidad de generalización y, por lo tanto, resulte más adecuado para su aplicación en escenarios reales.

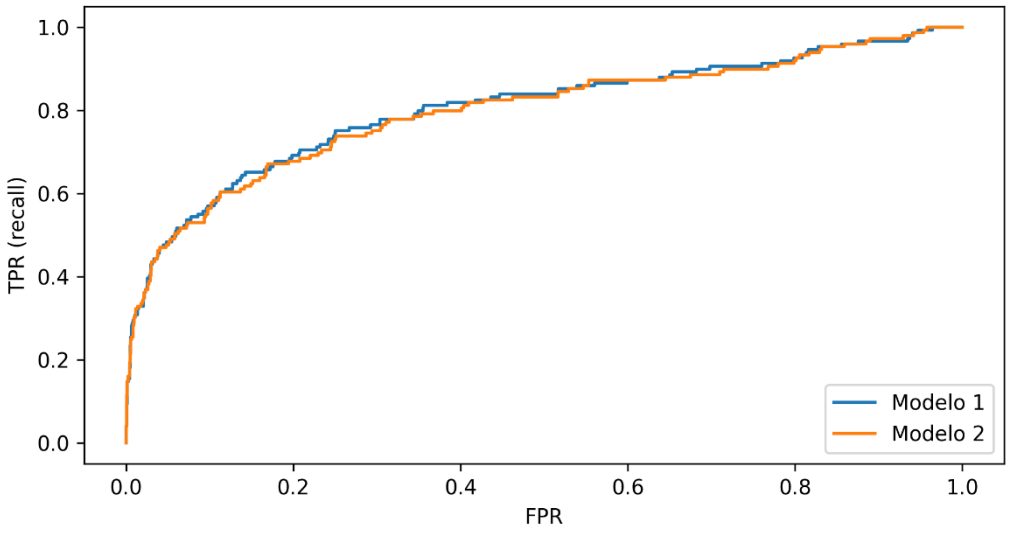
El tratamiento de los datos y la estimación de los modelos para cada algoritmo es llevado a cabo con el lenguaje Python mediante el uso de distintas librerías entre las que se encuentra *Pandas*, *Numpy* y *Scikit Learn*. En el anexo se encuentran los valores de los hiperparámetros óptimos obtenidos con la librería *Scikit Learn* para la estimación de los modelos a los datos de entrenamiento.

2.1. REGRESION LOGISTICA

La optimización mediante la búsqueda en cuadrícula por validación cruzada se lleva a cabo con $k = 5$ en ambos modelos, resultando en 800 posibles combinaciones y 4000 estimaciones por modelo. Los modelos óptimos presentan las siguientes curvas ROC:

$\{(0,1), (1,0), (1,1)\}$. Para cada una de estas combinaciones se estiman cinco modelos con diferentes bases de entrenamiento, obteniendo un total de 15 modelos. Se seleccionará aquel que maximice la métrica definida, *balanced accuracy* o AUC ROC.

Grafico 12: Curva ROC modelos Regresión Logística



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla 4: AUC ROC modelos Regresion Logística

	AUC ROC
Modelo 1	0.804
Modelo 2	0.798

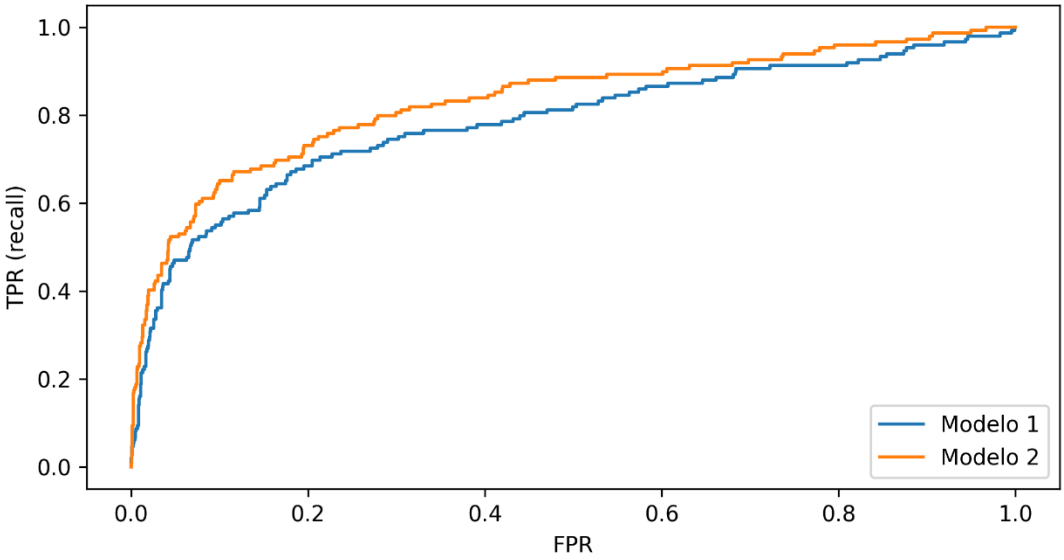
Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

El modelo 1 con sus hiperparámetros óptimos presenta un mejor desempeño predictivo en la base de prueba, por lo que se elige a este como el modelo de regresión logística óptimo.

2.2. SUPPORT VECTOR MACHINE

La optimización mediante la búsqueda en cuadrícula por validación cruzada se lleva a cabo con $k = 5$ en ambos modelos, resultando en 54 posibles combinaciones y 270 estimaciones por modelo. Los modelos óptimos presentan las siguientes curvas ROC:

Grafico 13: Curva ROC modelos SVM



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla 5: AUC ROC modelos Regresion Logística

	AUC ROC
Modelo 1	0.784
Modelo 2	0.833

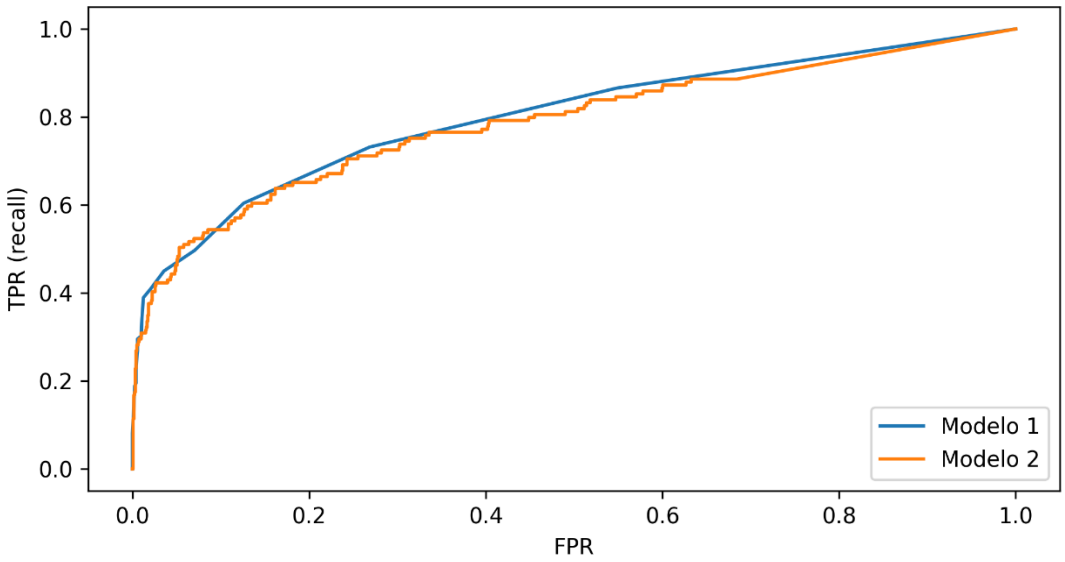
Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

El modelo 2 con sus hiperparámetros óptimos presenta un mejor desempeño predictivo en la base de prueba, por lo que se elige a este como el modelo Support Vector Machine óptimo. 0.784 y 0.833

2.3. K NEAREST NEIGHBOURS

La optimización mediante la búsqueda en cuadrícula por validación cruzada se lleva a cabo con $k = 5$ en ambos modelos, resultando en 232 posibles combinaciones y 1160 estimaciones por modelo. Los modelos óptimos presentan las siguientes curvas ROC:

Grafico 14: Curva ROC modelos KNN



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla 6: AUC ROC modelos KNN

	AUC ROC
Modelo 1	0.8
Modelo 2	0.787

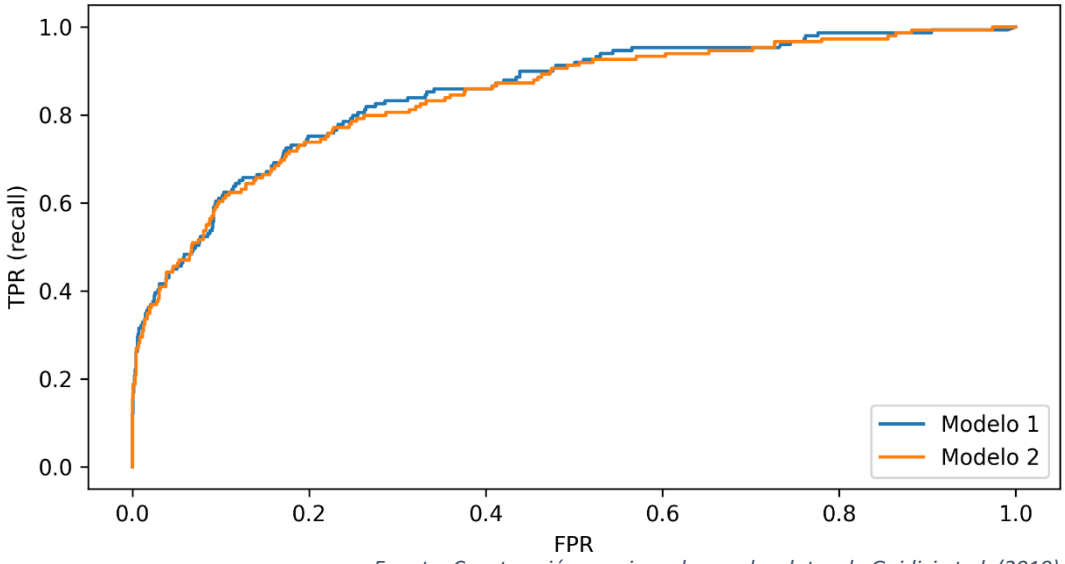
Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

El modelo 1 con sus hiperparámetros óptimos presenta un mejor desempeño predictivo en la base de prueba, por lo que se elige a este como el modelo K Nearest Neighbours óptimo.

2.4. RANDOM FOREST

La optimización mediante *GridSearchCV* se lleva a cabo con $k = 5$ en ambos modelos, resultando en 480 posibles combinaciones y 2400 estimaciones por modelo. Los modelos óptimos presentan las siguientes curvas ROC:

Grafico 15: Curva ROC modelos Random Forest



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla 7: AUC ROC modelos Random Forest

	AUC ROC
Modelo 1	0.850
Modelo 2	0.841

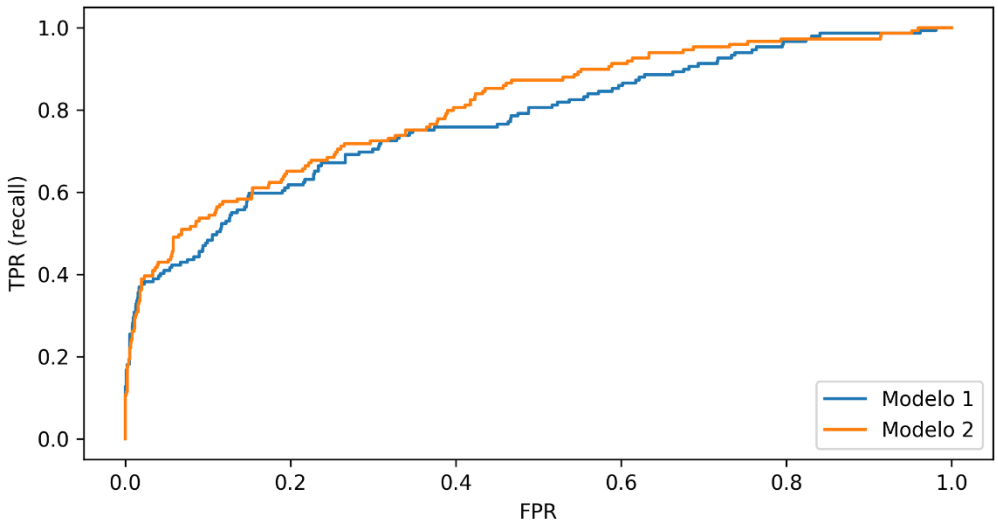
Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

El modelo 1 con sus hiperparámetros óptimos presenta un mejor desempeño predictivo en la base de prueba, por lo que se elige a este como el modelo Random Forest óptimo.

2.5. XGBOOST

La optimización mediante *GridSearchCV* se lleva a cabo con $k = 5$ en ambos modelos, resultando en 1458 posibles combinaciones y 7290 estimaciones por modelo. Los modelos óptimos presentan las siguientes curvas ROC:

Grafico 16: Curva ROC modelos XGBoost



Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla 8: AUC ROC modelos XGBoost

	AUC ROC
Modelo 1	0.824
Modelo 2	0.825

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

La estimación del modelo con los hiperparámetros del modelo 2 presenta un mejor desempeño predictivo en la base de prueba, por lo que se elige a este como el modelo XGBoost óptimo.

3. ELECCION ALGORITMO OPTIMO

En base a los resultados de las estimaciones de los modelos de clasificación optimizados se elige el modelo cuyo rendimiento predictivo sea superior, es decir, aquel modelo que presente la mejor capacidad de generalización, y de esa manera resulte más adecuado para su aplicación en escenarios reales.

A continuación, se presenta una tabla resumen con las métricas más utilizadas por la literatura actual en la evaluación de algoritmos de clasificación. Es importante tener en cuenta que todas estas, a excepción de AUC ROC, van a depender del umbral de clasificación por default de cada algoritmo, y por lo tanto, de la predicción específica que realicen según la base de prueba. Es por ello que las métricas se calculan para el umbral de clasificación óptimo de cada algoritmo obtenido como aquel umbral para el cual la diferencia entre la TPR y la FPR toma su valor máximo.

Tabla 9: Métricas de rendimiento predictivo para los modelos estimados

	Accuracy (1)	Recall (2)	Precision (3)	F1 Score (4)	Kappa (5)	AUC ROC (6)	Promedio (4) (5) (6)
Reg. Logística	0.83	0.65	0.36	0.46	0.38	0.803	61.2%
SVM	0.86	0.67	0.42	0.51	0.44	0.832	65.4%
KNN	0.84	0.6	0.37	0.46	0.38	0.799	61.0%
Random Forest	0.74	0.81	0.28	0.41	0.29	0.849	60.0%
XGBoost	0.79	0.71	0.31	0.43	0.33	0.825	60.3%

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Para el caso de clasificación binaria es importante identificar previamente a la elección del modelo óptimo cuál tipo de error es relativamente más costoso para el problema en cuestión. Si el error de tipo I es más costoso, entonces se buscará minimizar los Falsos Positivos que arroje la predicción, mientras que si el error de tipo II es más costoso se hará lo mismo con los Falsos Negativos. En general, para problemas de riesgo crediticio e identificación de potenciales clientes que incurran en incumplimiento de pago (default), el error de tipo II es relativamente más costoso ya que implica una pérdida cierta para el prestador, mientras que el error de tipo I se clasifica como costo de oportunidad o económico. Luego, en el presente trabajo resulta importante elegir aquel modelo que alcance el mejor equilibrio posible entre los tipos de errores teniendo en cuenta simultáneamente el mayor costo relativo del error de tipo II y la maximización de una posible función de ganancia empresarial. Por lo tanto, la decisión del algoritmo óptimo para el problema del presente trabajo dependerá de un promedio ponderado de AUC ROC, F1 Score y Kappa, con las cuales se obtiene un indicador equilibrado para el problema mencionado. Debido al mayor costo relativo del error de tipo II se pondera en un 50% el AUC ROC, dejando el restante 25% para el F1 Score y Kappa. Es importante aclarar que el resto de las métricas no son incluidas en el promedio ponderado debido a que ya se encuentran en el cálculo de las tres métricas elegidas.

De los resultados de la Tabla 9 se observa que el modelo SVM supera al resto en el balance total medido por el promedio ponderado. A pesar de no destacarse como el mejor algoritmo en

ninguna de las métricas individuales, sería el que mejor logra maximizar una función de ganancia asociada al problema estudiado.

Los modelos de la Tabla 9 son estimados 1000 veces cada uno con división de sets de entrenamiento y testo de manera aleatoria y así obtener la media de cada métrica y su correspondiente desvío estándar. Los resultados se muestran a continuación

Tabla 10: Media y desvío estandar para 1000 estimaciones de cada modelo

	Accuracy		Recall		Precision		F1 Score		Kappa		AUC ROC	
	Media	Std	Media	Std	Media	Std	Media	Std	Media	Std	Media	Std
Reg. Logística	0.847	0.032	0.57	0.05	0.385	0.06	0.454	0.039	0.371	0.051	0.766	0.021
SVM	0.802	0.044	0.687	0.06	0.328	0.055	0.439	0.042	0.34	0.055	0.807	0.018
KNN	0.813	0.052	0.625	0.071	0.342	0.074	0.433	0.047	0.337	0.064	0.788	0.019
Random Forest	0.796	0.042	0.727	0.057	0.325	0.051	0.445	0.041	0.345	0.053	0.834	0.016
XGBoost	0.8	0.038	0.73	0.052	0.329	0.047	0.449	0.038	0.35	0.049	0.836	0.016

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Mediante la media y el desvío estándar asociada a cada una de las métricas se concluye que dichos valores son consistentes.

CAPITULO III
APLICACION: CASO AFLUENTA

Resulta interesante llevar a cabo un caso de aplicación con datos de la realidad a partir del modelo óptimo obtenido en el capítulo II. Por lo tanto, la idea central de este capítulo es la de simular el uso del modelo SVM por parte de una plataforma P2P para la decisión de otorgar o no el préstamo solicitado por el cliente teniendo en cuenta los costos y beneficios asociados a la operatoria de la misma, y de esa manera obtener una aproximación a la función de beneficios de esta.

Es importante tener en cuenta que el presente caso de aplicación tiene como fin la simulación del uso del modelo óptimo para la clasificación de clientes basándose en datos provistos por la plataforma P2P descripta en el capítulo I (Afluenta), a través de modelos estimados a partir de datos de PyMES italianas, por lo que los resultados del ejercicio deben ser interpretados con cierta precaución. No obstante, el objetivo del ejercicio es mostrar la potencial utilidad de los algoritmos empleados en la clasificación de clientes. La precisa estimación de los beneficios se configura como una posible línea de investigación futura.

Para llevar a cabo este caso de aplicación se realizó una serie de supuestos con relación a la forma operativa de estas plataformas que permiten simplificar el análisis en cuestión. Los mismos son definidos de manera tal de acercarse a un escenario promedio en la operatoria dentro de la plataforma:

- Afluenta cobra la misma comisión a todos los clientes.
- El plazo de devolución del crédito solicitado es de 12 meses.
- Se tiene en consideración únicamente el crédito cuyo monto es de $2.000.000.
- No existe costo de procesamiento de la solicitud de crédito.

Para la simulación es necesario contar con datos relacionados a los costos e ingresos que surjan de la operatoria por parte de la plataforma. En este sentido, es importante también identificar los conceptos por los que surgen dichos costos e ingresos. Por lo tanto, a partir de datos públicos de Afluenta se obtienen los siguientes datos:

- Ingresos potenciales: 6,23% de comisión sobre el monto solicitado.
- Costos potenciales: 42,35% sobre el saldo a pagar por gestión de mora y gestión judicial.

En este punto, resulta importante destacar una particularidad de las plataformas P2P: los costos ciertos están asociados a gastos por gestión de mora y judicial ante un caso de *default*, y no incluyen costos por el lado de la pérdida de capital prestado. Esto se debe a que, como se mencionó anteriormente, estas plataformas no asumen responsabilidad o riesgo alguno por las operaciones entre inversores y tomadores de créditos ya que no garantiza el cobro de éstos, limitándose a realizar gestiones para que los deudores cumplan con sus obligaciones. En relación con este punto, es interesante observar cual será la estrategia óptima para llevar a cabo por parte de las plataformas debido a los incentivos que generan la no internalización del riesgo operacional.

Para la estimación de la posible función de beneficios a partir de la simulación es necesario identificar los costos y beneficios asociados a cada resultado posible de predicción. Para ello, es útil trabajar con la matriz de confusión descripta en el Capítulo I, donde se podrá asignar un costo o beneficio a cada elemento de esta.

Tabla 11: Matriz de confusión para la predicción del modelo SVM

		Observación	
		1	0
Predicción	1	100 (TP)	140 (FP)
	0	49 (FN)	1066 (TN)

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Los costos y beneficios asociados a cada uno de los elementos de la matriz de confusión y en base al monto del crédito utilizado para el análisis son los siguientes:

- **Costo del Falso Positivo (FP) – (\$126.630):** Comisiones anuales por haber otorgado el préstamo a un cliente que hubiera pagado.
- **Costo del Falso Negativo (FN) - (\$847.700):** Pérdida cierta por la gestión de mora y gestión judicial a causa del préstamo a un cliente que no cumplió con el pago de este.
- **Costo del Verdadero Positivo (TP) – (\$0):** Costo del procesamiento de la solicitud.
- **Beneficio del Verdadero Negativo (TN) – (\$126.630):** Comisión anual ganada por el cliente que paga.

Luego, a partir de la predicción llevada a cabo por el modelo y los datos asociados a cada una de estas predicciones se obtiene el valor de \$73.904.380 de beneficio para Afluenta si utilizara este modelo para la clasificación de sus clientes.

Como se mencionó anteriormente, el objetivo final de esta aplicación es la de simular el uso del modelo en la clasificación de clientes, por lo que el valor nominal del beneficio atribuible a la plataforma por el uso del algoritmo SVM no es de real importancia, sino el signo de este (beneficio o costo) y la comparación respecto al uso de otros algoritmos, y en particular respecto a la estrategia de maximización de transacciones que podría llevar a cabo la plataforma (para la cual todos los clientes serían clasificados como 0). En la siguiente tabla se resume dicha información.

Tabla 12: Cuadro resumen de beneficios

ALGORITMO	GANANCIA	RELATIVO A SVM
SVM	\$ 73,904,380	-
Reg. Logística	\$ 63,387,060	-17%
KNN	\$ 62,443,260	-18%
XGBoost	\$ 54,309,640	-36%
Random Forest	\$ 47,920,840	-54%
Maximizador de cantidad (prediccion todos 0)	\$ 24,100,780	-207%

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

De la Tabla 12 es notable cómo el mejor modelo de clasificación considerado genera una ganancia relativa mayor; en particular con relación a la estrategia maximizadora de transacciones. Es decir, a pesar de la baja probabilidad de ocurrencia de un default los mayores ingresos asociados a una cantidad nula de falsos positivos no compensan las pérdidas generadas por todos los falsos

negativos presentes, por lo que se verifica el mayor costo relativo de los falsos negativos y se demuestra que esta estrategia que surge a partir de la no internalización del riesgo de capital por parte de las plataformas P2P no es óptima desde el punto de vistas privado.

Profundizando un poco más en el análisis, podemos llegar a la conclusión de que la adopción del modelo de clasificación óptimo por parte de estas plataformas implica también un beneficio desde el punto de vista social debido a la menor cantidad de transacciones que terminan en default, beneficiando en este caso a los inversores.

CONCLUSIONES

Como se mencionó anteriormente, el principal objetivo de este trabajo de investigación es la determinación del algoritmo óptimo para la clasificación de clientes en pos de minimizar el riesgo asociado a la operatoria propia de las plataformas P2P. En este sentido, para los algoritmos considerados, el modelo estimado a partir del algoritmo SVM fue superior en rendimiento predictivo global medido por diversas métricas. Es decir, el modelo SVM es el que mejor balance obtiene entre falsos positivos y falsos negativos.

En relación con el segundo objetivo del trabajo, la respuesta deriva de la primera conclusión quedando probado que el modelo Logit es superado en términos predictivos por el modelo SVM, en línea con lo que se observa en la literatura actual. Sin embargo, es importante destacar que las métricas utilizadas en el presente trabajo para llegar a dicha conclusión se enfocan únicamente en el aspecto predictivo del modelo, y no tienen en cuenta atributos como la interpretabilidad de este. En este sentido, es importante destacar que el modelo Logit ofrece una alta interpretabilidad mientras que el modelo SVM no ofrece ninguna y, por lo tanto, se deriva de ello una posible línea de investigación que incorpore en el análisis comparativo de algoritmos métricas que permitan cuantificar dichos atributos y así lograr obtener un modelo óptimo a partir del *trade-off* entre predicción e interpretabilidad.

Finalmente, se obtuvieron una serie de conclusiones a partir del caso de aplicación propuesto que configuran posibles líneas de investigación. Por un lado, teniendo en cuenta el diseño particular de las plataformas P2P y los incentivos que ello genera en relación con la maximización de beneficios, es a partir de una simple aplicación con el uso de la muestra de prueba que se base en los datos de una plataforma P2P de la realidad como Afluenta que se concluye que dicha maximización de beneficios tiene como estrategia óptima la del uso de modelos de puntaje para la clasificación de clientes, en detrimento de la estrategia de maximización de transacciones. Una posible línea de investigación podría implicar otros enfoques en el tipo de aplicación para profundizar en este análisis, mejorando la precisión en las estimaciones de costos/beneficios con utilizando datos argentinos y menos supuestos. Por otro lado, se observa que mientras mejor sea el desempeño predictivo del modelo elegido no se beneficia únicamente la plataforma P2P, sino que también se ven beneficiados los inversores a través de una menor cantidad de transacciones que terminan en default, configurando esto un beneficio social atribuible al uso de algoritmos de

clasificación. En este sentido, resulta interesante profundizar en la cuantificación de dicho beneficio social y que sirva para la evaluación de posibles regulaciones a las plataformas P2P.

BIBLIOGRAFÍA

Ahumada, H. A., Gabrielli, M. F., Herrera Gomez, M. H., & Sosa Escudero, W. (2018). Una nueva econometría: Automatización, big data, econometría espacial y estructural. Bahía Blanca: Editorial de la Universidad Nacional del Sur.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., y Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54:627–635.

Bussmann, N., Giudici, P., Marinelli, D., y Papenbrock, J. (2021). Explainable machine learning in credit risk management. *Computational Economics*, 57(1), 203-216.

Coffman, J. (1986). The proper role of tree analysis in forecasting the risk behavior of borrowers. *Management Decision Systems, Atlanta, MDS Reports*, 3(4):7.

Dumitrescu, E.I., Hué, S., Hurlin, C., y Tokpavi, S. (2021). Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds. Working Paper. Recuperado de SSRN: https://ssrn.com/abstract=3553781

Dumrauf, G. (2010). Finanzas Corporativas: un enfoque latinoamericano. Buenos Aires: Alfaomega Grupo Editor Argentino.

Desai, V. S., Crook, J. N., y Overstreet Jr, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, 95(1):24–37.

Difabio de Anglat, G. Apuntes de cátedra Metodología de la Investigación. Bloque temático N°1: Diseño del objeto de investigación. Doctorado en Ciencias Económicas, Facultad de Ciencias Económicas, Universidad Nacional de Cuyo. 2021

Ekong, R. E., Akintola, K. G., y Kuboye, B. M. (2022). Development Of Credit Scoring Model For Borrowers Using Machine Learning Techniques. *PERSPEKTIF*, 11(3), 829-838.

Giudici, P., Hadji-Misheva, B., y Spelta, A. (2019). Network based scoring models to improve credit risk management in peer to peer lending platforms. *Frontiers in artificial intelligence*, 2(3), 1-8.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

Halim, S., y Humira, Y. V. (2014). Credit Scoring Modeling. *Journal Teknik Industri*, 1(1), 17-24.

Henley, W. y Hand, D. (1996). A k-nearest-neighbour classifier for assessing consumer credit risk. *The Statistician*, 45(1):77–95.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2018). Metodología de la investigación. México: McGraw-Hill Interamericana.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. New York: Springer.

King, B. (2010). Bank 2.0: How customer behaviour and technology will change the future of financial services. Marshall Cavendish International Asia Pte Ltd.

Ko, P. C., Lin, P. C., Do, H. T., y Huang, Y. F. (2022). P2P Lending Default Prediction Based on AI and Statistical Models. *Entropy*, 24(6), 1-23.

Lessmann, S., Baesens, B., Seow, H.-V., y Thomas, L. C. (2015). Benchmarking stateof-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research,* 247:124–136.

Makowski, P. (1985). Credit scoring branches out. *Credit World*, 75(1):30–37.

Mach, T. L., Carter, C. M., y Slattery, C. R. (2014). Peer-to-peer lending to small businesses. Divisions of Research y Statistics and Monetary Affairs Federal Reserve Board. Working Paper No. 2014-10. Washington D.C., WA.

Moenninghoff, S. C., & Wieandt, A. (2013). The future of peer-to-peer finance. *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung*, 65(1): 466-487.

Srinivasan, V. y Kim, Y. H. (1987). Credit granting: A comparative analysis of classification procedures. *The Journal of Finance*, 42(3):665–681.

Tam, K. Y. y Kiang, M. Y. (1992). Managerial applications of neural networks: the case of bank failure predictions. *Management Science*, 38(7):926–947.

Vinay, S. (2021). Standardization in Machine Learning. LinkedIn. Recuperado de https://www.linkedin.com/pulse/standardization-machine-learning-sachin-vinay/.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, 27(11-12):1131–1152.

XGBoost developers (2022). Xgboost Documentation. Revision 36eb41c9. Recuperado de https://xgboost.readthedocs.io/en/stable/.

Yobas, M. B., Crook, J. N., y Ross, P. (2000). Credit scoring using neural and evolutionary techniques. IMA *Journal of Mathematics Applied in Business and Industry*, 11:111–125.

ANEXO I
ESTIMACIÓN DE MODELOS

La estimación de los modelos de aprendizaje automático fue realizada en Python con el uso de la libreria *Scikit Learn*. A continuación, se presentan los hiperparámetros optimizados de cada modelo y su valor óptimo

Tabla A.1: Hiperparámetros óptimos regresion logística

	HIPERPARAMETROS			
	Penalty	**C**	**Solver**	**Class_weight**
Modelo 1	L2	0.234	Liblinear	Balanced
Modelo 2	L2	0.002	Liblinear	Balanced

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla A.2: Hiperparámetros óptimos SVM

	HIPERPARAMETROS				
	Kernel	**Gamma**	**Degree**	**C**	**Class_weight**
Modelo 1	Poly	-	2	10	Balanced
Modelo 2	RBF	1	-	0.01	Balanced

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla A.3: Hiperparámetros óptimos KNN

	HIPERPARAMETROS		
	N_Neighbours	**Weights**	**Metric**
Modelo 1	19	Uniform	Manhattan
Modelo 2	30	Distance	Minkowski

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla A.4: Hiperparámetros óptimos Random Forest

	HIPERPARAMETROS			
	N_Estimators	**Min_Samples_Split**	**Min_Samples_Leaf**	**Criterion**
Modelo 1	50	2	5	Gini
Modelo 2	50	15	5	Entropy

Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

Tabla A.5: Hiperparámetros óptimos XGBoost

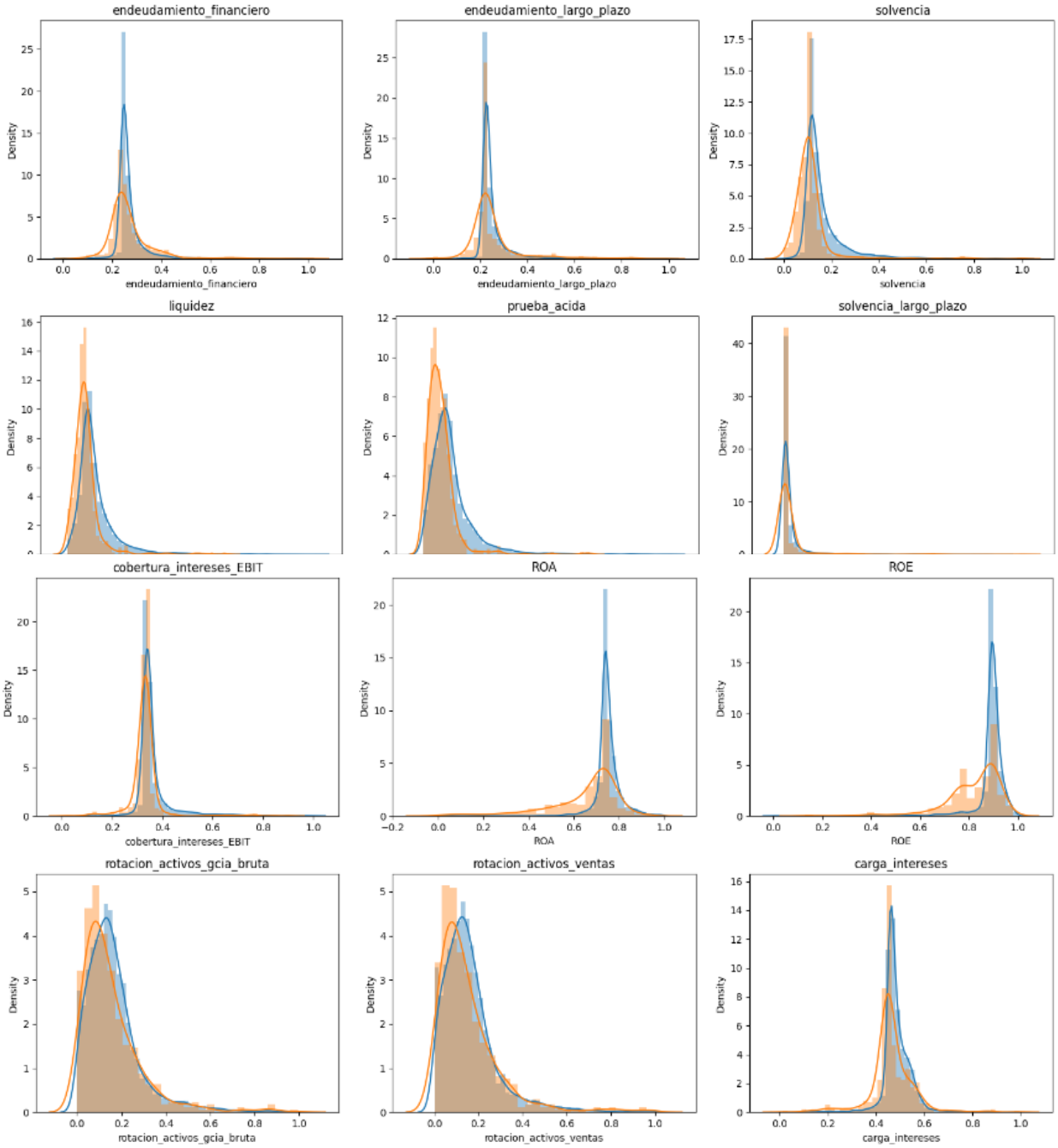
	HIPERPARAMETROS					
	N_Estimators	**Min_Child_Weight**	**Subsample**	**Colsample_bytree**	**Max_depth**	**Scale_pos_weight**
Modelo 1	50	10	0.8	0.6	4	5
Modelo 2	50	15	0.8	1	3	1

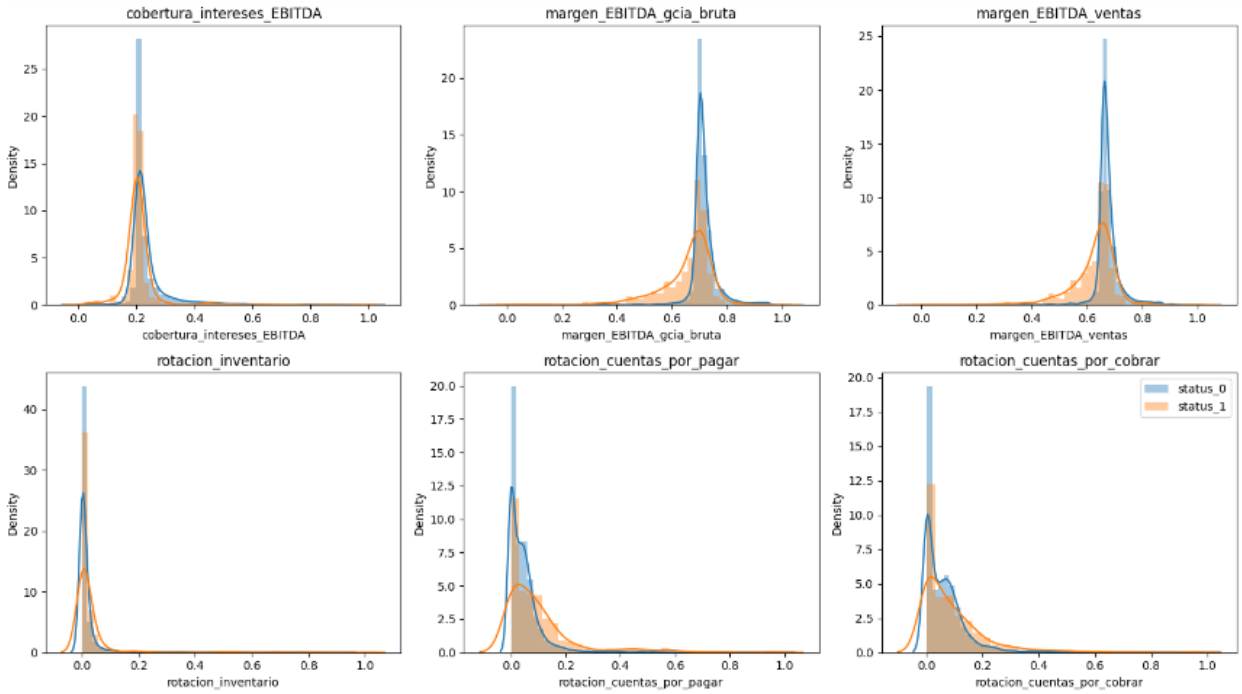
Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

ANEXO II
DISTRIBUCIÓN DE LOS RATIOS FINANCIEROS

A continuación, se presentan las distribuciones de los ratios financieros utilizados en el trabajo luego de ser escalados mediante *min-max scaler*.

Grafico A.1: Distribución de ratios financieros





Fuente: Construcción propia en base a los datos de Guidici et al. (2019)

ANEXO III
GLOSARIO DE TÉRMINOS PROVENIENTES DEL INGLÉS

- *FinTechs:* Combinación de las palabras “Finanzas” y “Tecnología” y el término se refiere a empresas que utilizan la tecnología para proporcionar servicios financieros de manera innovadora en desafío a métodos tradicionales.

- *Peer to peer lending:* Se traduce como préstamo entre pares y hace referencia a al modelo de préstamos facilitados entre individuos a través de una plataforma en línea.

- *Machine Learning:* Conjunto de algoritmos cuya finalidad es predecir un cierto resultado en base a un conjunto de datos dado.

- *Bootstraping:* Es una técnica que se utiliza para estimar la variabilidad de un modelo de aprendizaje automático. La técnica implica la creación de múltiples conjuntos de datos de entrenamiento a partir del conjunto original utilizando un muestreo con reemplazo.

- *Boosting:* Técnica de ensamblaje de modelos que se utiliza para mejorar el rendimiento de los modelos de aprendizaje automático y reducir el sesgo y la varianza.

- *Blockchain:* Tecnología de registro distribuido que permite el almacenamiento y la trasnferencia segura de datos de manera descentralizada.

- *Start up:* Es una compañía que esta en sus etapas iniciales de desarrollo.

- *European External Credit Assessment Institutions (ECAI):* Instituciones calificadoras crediticias europeas.

- *Spread:* Se refiere a la diferencia entre las tasas de interés de dos instrumentos financieros o activos distintos

DECLARACIÓN JURADA RESOLUCIÓN 212/99 CD

El autor de este trabajo declara que fue elaborado sin utilizar ningún otro material que no haya dado a conocer en las referencias que nunca fue presentado para su evaluación en carreras universitarias y que no transgrede o afecta los derechos de terceros.

Mendoza, 27 DE JULIO DE 2023

GARCIA OJEDA, JUAN GABRIEL

Firma y aclaración

31.105

Número de registro

42.168.367

DNI