AFI Escuela de Finanzas

Máster en Data Science y Big Data en Finanzas

TRABAJO FIN DE MÁSTER

Interpretabilidad, Machine Learning y Riesgo de Crédito

por

Gustavo Eduardo Vargas Núñez

Agosto 2019

Tutor: Miguel Ángel Corella

Agradecimientos

A Dora, Elen y mis amigos.

Resumen

En los últimos años, la llegada del Machine Learning ha afectado a la forma en que las entidades financieras hacen las admisiones de solicitudes de préstamo. El problema reside en que estos modelos se consideran modelos *de caja negra*, esto es, que no podemos saber los motivos de determinada clasficación. Para atajar este problema, en los últimos años se han estado desarrollando paquetes de interpretabilidad para modelos de Machine Learning, cuya finalidad es hacer interpretables esas *cajas negas*.

En este trabajo usamos los paquetes **eli5**, **Lime** y **shap** para conseguir explicaciones de nuestros modelos. Los aplicamos sobre la base de datos que proporciona la empresa de préstamos LendingClub en un problema de credit *scoring*.

Keywords : Credit Risk, Machine Learning, Interpretabilidad, Lime, Shap, Eli5

Índice

Índice de figuras

1. Introducción

A mediados de los 70 se eliminó la convertibilidad de dinero en oro y con ello se terminó de implantar el uso de monedas fiduciarias a nivel mundial. Esto provocó que aumentase la incertidumbre sobre la inflación, el tipo de cambio y el tipo de interés. Prueba de ello es el gran aumento de gasto sobre PIB que desde entonces se hace en intermediarios y asesores financieros para que gestionen este mayor riesgo [1].

Uno de los riesgos que se ha puesto de mayor relieve en las últimas crisis financieras es el riesgo de crédito [2]. Este riesgo se define como "el riesgo de pérdida financiera producida por el incumplimiento o deterioro de la calidad crediticia de un cliente o un tercero, al cual una entidad financiera ha financiado o por el cual se ha asumido una obligación contractual" [3]. En los últimos tiempos se han comenzado a utilizar los modelos de machine learning para el cálculo del credit scoring, esto es, para el cálculo del riesgo de incumplimiento por impago de ese crédito.

Una característica deseable para estos modelos es que se puedan explicar e interpretar. Estos métodos se caracaterizan por funcionar como *cajas negras*, por lo que se prefiere usar métodos más simples, como la regresión lineal o la regresión logística. En el presente trabajo pretendemos ver hasta qué puntos los paquetes actuales de interpretabilidad permiten entender las decisiones que toma un determinado método de Machine Learning, aplicado a un problema de credit scoring.

Como veremos, si bien no alcanzamos una interpretabilidad completa, sí obtenemos una interpretabilidad *local* en los distintos métodos.

1.1. Motivación de la investigación

Con lo anterior presente, planteamo la pregunta que define nuestra investigación:

- ¿Hasta qué punto son interpretables los modelos de Machine Learning?

1.2. Contribuciones

En este trabajo hemos hecho lo siguiente:

- Por un lado, hemos calculado el Information Value sobre nuestro dataset. Esta técnica intenta hallar las variables más significativas de cara a predecir nuestro target, para posteriormente usarlas en modelos de predicción interpretables.
- Por otro lado, con los paquetes de interpretabilidad eli5, Lime y Shap, obtenemos explicaciones locales de cómo funcionan nuestros modelos de Machine Learning.

1.3. Estructura

En el apartado 2 desarrollo los fundamentos de los modelos de interpretabilidad y de los métodos de Machine Learning que usaremos. También explicamos en qué consiste el Weight of Evidence/Information Value. En el apartado 3 comentamos los experimentos que realizamos sobre nuestro dataset, LendingClub. En el apartado 4 tenemos nuestras conclusiones y futura investigación sobre los modelos de interpretabilidad. Por último, en los anexos tenemos más

detalles tanto sobre el preprocesado que hemos hecho a los datos como sobre el proceso de optimización de nuestros hiperparámetros.

2. Estado del arte

En los siguientes puntos comentamos en qué consisten los métodos y modelos usados en este trabajo.

2.1. En reducción de dimensionalidad en riesgo de crédito

2.1.1. Weight of Evidence\Information Value

El *Weigth of Evidence* y el *Information Value* son técnicas simples para la selección y reducción de variables. La analizamos aquí porque se suele usar a la hora de tratar datos para hacer credit scoring[4].

En general, usamos WOE al tener una variable target binaria, donde buscaremos clasificar cada observación en buena\mala o evento\no evento. El WOE separa cada variable por categorías, en el caso de variables categóricas, o bien crea grupos (*binning*) en el caso de una variable numérica continua, y calcula la relación entre el número total de eventos de esa categoría frente al total de observaciones de esa categoría. Hace lo mismo con los "no eventos" y ejecuta las fórmulas que veremos a continuación.

El WOE se define como:

$$WOE = ln\left(\frac{Event\,\%}{Non\_event\,\%}\right) \tag{1}$$

Y el Information Value como:

$$IV = \sum_{i=1}^{n}\left((Event\,\% - Non\_event\,\%) * ln(\frac{Event\,\%}{Non\_event\,\%})\right) \tag{2}$$

o simplemente:

$$IV = \sum_{i=1}^{n}((Event\,\% - Non\_event\,\%) * WOE) \tag{3}$$

Las ventajas de usar esta técnica[5] es que trata muy bien los missing values y los outliers, no necesita crear variables dummy y usar relaciones logarítmica, lo que lo hace ideal para una selección de variables que luego vayan a usarse en una regresión logística.

Una vez obtenemos el Information Value de cada variable, podemos seleccionarlas según el siguiente criterio que vemos en el Cuadro 1.

Information Value	Poder predictivo
<0.02	inútil para predecir
0.02 - 0.1	predictor débil
0.1 - 0.3	predictor medio
0.3 - 0.5	predictor fuerte
>0.5	demasiado bueno, verificar

Cuadro 1: Poder predictivo del Information Value

2.2. En las técnicas de predicción

A continuación explicamos los métodos de Machine Learning que usaremos con nuestro dataset. Hemos escogido la regresión logística por ser un método ampliamente utilizado en el sector de riesgo de crédito, y los otros métodos por ser los más usados dentro del ámbito del Machine Learning y/o compatibles con el paquete Lime.

2.2.1. Regresión logística

También llamada sigmoide por su forma de s, la regresión logística ha sido una de las más importantes y usadas en problemas de clasificación. Su simplicidad y poder predictivo hacen que haya sido una herramienta elegida por muchos. También ayuda el hecho de que tenga una derivada sencilla de calcular y los valores que toma están limitados entre cero y uno.

Para crear probabilidades, pasamos una función z a través de la sigmoide $\sigma(z)$, que viene dada por la siguiente función:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

Es posible usar una versión multinomial de la regresión logística, llamada *softmax* para clasificar más de dos clases, pero en este trabajo no es necesario.

2.2.2. Árboles de decisión

La regresión logística y lineal no funcionan cuando la relación entre las variables y el target no es lineal o cuando las variables están relacionadas entre ellas. Aquí es donde los árboles de decisión, ya que van dividiendo los datos hasta cierto valor límite de las variables. Al hacer divisiones va creando subconjuntos a los que pertenecen los datos. Los subconjuntos finales se llaman nodos terminales o hojas, y los intermedios se llaman nodo no terminal o interno. Para predecir el output en un determinada hoja, se calcula la media de datos de entramiento para esa hoja.

El algoritmo que viene implementado en el paquete sklearn, que es el que usaremos en los experimentos, es el algoritmo CART, aunque hay más alternativas.

2.2.3. Random Forest

El problema de los árboles de decisión es que pueden ser muy sensibles a cambios en los datos de entrenamiento. Por ello, lo que hace Random Forest es entrenar muchos árboles, cada uno con un subconjunto aleatorio con reemplazo, o *bagging*, con los datos.

2.2.4. Extreme Gradient Boosting

Gradient Boosting es una versión del Boosting donde se establece una función de pérdida, que optimizaremos mediante descenso por gradiente. De facto, el gradiente nos lleva a buscar la combinación óptima de los ensembles que crea el Boosting.

Extreme Gradient Boosting es un caso particular del anterior, solo que aplica regularización a la función de pérdida y está muy especializado en construir ensembles de árboles. Otras características de este método de optimización es que tiene poda integrada y proporciona los

valores de las hojas.

2.3. En las técnicas de interpretabilidad

2.3.1. Eli5

ELI5 (Explain me Like I'm 5) es un paquete que nos ayuda a interpretar algoritmo de predicción y clasificación de Machine Learning. Es el más sencillo de los tres paquetes que vamos a ver aquí. Si bien en su origen solo podía explicar modelos ya de por sí interpretables o de *caja blanca*, en posteriores actualizaciones ha comenzado a ser capaz de explicar otro tipo de modelos. Este paquete sigue dando pasos para que sus explicaciones sean agnóticas del modelo; y por ahora ya puede explicar métodos basados en árboles.

Puede usarse para dar una explicación global en determinados modelos, y también para explicar el proceso de predicción para determinada observación. Toda la documentación sobre este paquete se puede ver aquí.

2.3.2. Lime

El paquete LIME (Local Interpretable Model-agnostic Explanations) pretende explicar el comportamiento de cualquier método de Machine Learning usando explicaciones *locales*. Para ello:

- Genera datos cercanos a la observación que queremos explicar
- Mete como input esos datos sintéticos por el método de caja negra que queremos explicar, obteniendo un output
- Finalmente, adecúa un modelo que sí sea interpretable, como una regresión lineal o un árbol de decisión, al input y output de antes, ponderando los datos sintéticos según su proximidad a la observación.

Para el caso de variables tabulares, este paquete requiere que los datos estén sin haber hecho one hot encoding, mientras que los modelos entrenados sí necesitan de ese paso en el preprocesado. Por ello, son necesarias ambas versiones de los datos. La conexión con sklearn está optimizada en este sentido, pero la conexión con keras es un tanto más complicada[1], por lo que en la parte de experimentos no hemos podido presentar la interpretabilidad de una red neuronal.

2.3.3. Shap

El último modelo de interpretabilida que probamos es SHAP (SHapley Additive exPlanations). SHAP usa tanto la teoría de juegos como las explicaciones locales. Para una predicción en particular, Shap asigna a cada variable un valor llamado Shap Value:

$$\phi_{ij} = \sum_{todos.los.ordenes} val(\{variables.antes.de.j\} \cup x_{ij}) - val(\{variables.antes.de.j\}) \quad (5)$$

[1]En concreto, la funcionalidad predict_proba de sklearn devuelve la probabilidad de pertenecer y no pertenecer a una clase. Sin embargo, los modelos de Keras, en su forma funcional, no ofrecen ese *predict_proba* y la forma Secuencial solo devuelve la probabilidad de pertencer a la clase, por lo que Lime lo rechaza.

Los shap values intentan explicar el output del modelo como la suma de los efectos que cada variable introduce, pero de manera condicional a qué variables han afectado antes. Los shap values son el resultado de hacer la media del efecto de todos los posibles órdenes de entrada. Desde la teoría de juegos se puede probar que este es el único enfoque consistente.

Aunque Shap puede explicar cualquier método de Machine Learning, es muy lento haciéndolo, especialmente si los modelos son complejos. Aún así, Shap por debajo tiene su código optimizado usando C++ para modelos basados en árboles.

3. Experimentos

Con los datos del dataset LendingClub, establecemos un problema de credit scoring. Necesitamos predecir dos clases, los que van a pagar (*Fully Paid*) y los que van a presentar algún problema con el pago (*Default*). Para más detalles, ver el Apéndice A. Y para replicar los experimentos, ver el repositorio en github.

En los métodos de Machine Learning, además de presentar lo ofrecido por los métodos de interpretabilidad, también presentaremos las medidas de *Precision*, que es la proporción de correctamente predichos para una clase dentro de todos los predichos de esa clase, y *Recall*, que es la cantidad de correctamente predichos de una clase dentro de todos los que realmente pertenecen a esa clase.

3.1. Variables más importantes para la regresión logística

Para el dato $x\_test = 5$, podemos ver las variables más importantes de la regresión logística según el IV, eli5 y Lime en el Cuadro 2.

IV	Eli5	Lime
issue_d	total_pymnt	total_pymnt
out_prncp	total_rec_prncp	total_rec_prncp
int_rate	total_rec_int	recoveries
sub_grade	loan_amnt	total_rec_int
total_rec_late_fee	out_prncp	issue_d

Cuadro 2: Variables importantes para la regresión logística

3.2. Variables más importantes para el árbol de decisión

Volvemos a analizar en $x\_test = 5$ la importancia de las variables, en este caso según el método de los árboles de decisión, en el Cuadro 3.

IV	Eli5	Lime
issue_d	last_pymnt_d	last_pymnt_d
out_prncp	total_rec_late_fee	total_rec_late_fee
int_rate		last_pymnt_amnt
sub_grade		total_rec_prncp
total_rec_late_fee		total_pymnt

Cuadro 3: Variables importantes para el árbol de decisión

3.3. Variables más importantes para Random Forest

Volvemos a analizar $x\_test = 5$.

IV	Eli5	Lime
issue_d	last_pymnt_d	recoveries
out_prncp	out_prncp	last_pymnt_d
int_rate	recoveries	collection_recovery_fee
sub_grade	collection_recovery_fee	total_rec_late_fee
total_rec_late_fee	total_rec_late_fee	total_rec_prncp

Cuadro 4: Variables importantes para random forest

3.4. Variables más importantes para Extreme Gradient Boosting

Volvemos a analizar $x\_test = 5$.

IV	Eli5	Lime	Shap
issue_d	term	recoveries	issue_d
out_prncp	total_rec_prncp	last_pymnt_d	last_pymnt_d
int_rate	last_pymnt_d	total_rec_prncp	term
sub_grade	recoveries	issue_d	out_prncp
total_rec_late_fee	total_rec_late_fee	out_prncp	loan_amnt

Cuadro 5: Variables importantes para XGBoost

3.5. Variables más importantes en observaciones mal predichas

Vamos a realizar el mismo análisis pero, en esta ocasión, solo con el paquete Lime sobre dos casos distintos donde ha habido un error en la predicción

3.5.1. Fully paid predicho como Default

En este caso analizamos $x\_test = 12$, que se ha predicho como Default pero su verdadero label es Fully Paid.

Regresión Logística	Árboles de decisión	Random Forest	XGBoost
recoveries	last_pymnt_d	last_pymnt_d	recoveries
total_pymnt	total_rec_late_fee	recoveries	last_pymnt_d
total_rec_prncp	last_pymnt_amnt	collection_recovery_fee	out_prncp
term	recoveries	total_rec_prncp	term
verification_status	dti	int_rate	issue_d

Cuadro 6: Variables importantes según Lime para x_test=12

3.5.2. Default predicho como Fully Paid

En este caso analizamos $x\_test = 1487$, que se ha predicho como Fully Paid pero su verdadero label es Default.

Regresión Logística	Árboles de decisión	Random Forest	XGBoost
total_pymnt	total_rec_late_fee	recoveries	recoveries
total_rec_prncp	last_pymnt_d	last_pymnt_d	last_pymnt_d
recoveries	total_rec_prncp	collection_recovery_fee	out_prncp
total_rec_int	last_pymnt_amnt	total_rec_prncp	issue_d
total_rec_late_fee	collection_recovery_fee	last_pymnt_amnt	total_rec_prncp

Cuadro 7: Variables importantes según Lime para x_test=1487

4. Conclusiones y futura investigación

4.1. Conclusiones

Como hemos podido ver, aunque la importancia de las explicaciones locales convergen, no terminan de hacerlo del todo, de lo que podemos deducir que una explicación *global* puede no existir. Lo anterior nos dice que el modelo solo puede ser *interpretable* localmente. Los métodos de Machine Learning son modelos fuertemente no lineales, por lo que cualquier explicación lineal no será certera, pero una convergencia nos aseguraría, al menos, que estamos en el camino correcto.

4.2. Futura investigación

Los nuevos trabajos sobre interpretabilidad apuntan a mejorar paquetes como Shap, ya que permite ver hasta qué punto una explicación local será consistente. Por otro lado, es necesario ver más explicaciones en observaciones donde los modelos fallan, ya que eso puede permitir mejorar los modelos o entender qué ven y qué no ven los modelos y qué tienen de especial los datos.

A. Dataset

A.1. Features del dataset Lendingclub

Los datos se han obtenido de esta página de Kaggle, en su versión 1. El listado y explicación de todas las features se pueden ver en el cuadro 8.

Cuadro 8: Features del dataset LendingClub

Índice	Nombre de la columna	Descripción
0	id	A unique LC assigned ID for the loan listing
1	member_id	A unique LC assigned Id for the borrower member
2	loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan_amount, then it will be reflected in this value
3	funded_amnt	The total amount committed to that loan at that point in time
4	funded_amnt_inv	The total amount committed by investors for that loan at that point in time
5	term	The number of payments on the loan. Values are in months and can be either 36 or 60
6	int_rate	Interest Rate on the loan
7	installment	The monthly payment owed by the borrower if the loan originates
8	grade	LC assigned loan grade
9	sub_grade	LC assigned loan subgrade
10	emp_title	The job title supplied by the Borrower when applying for the loan
11	emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years
12	home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER
13	annual_inc	The self-reported annual income provided by the borrower during registration
14	verification_status	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
15	issue_d	The month which the loan was funded
16	loan_status	Current status of the loan
17	pymnt_plan	Indicates if a payment plan has been put in place for the loan
18	url	URL for the LC page with listing data
19	desc	Loan description provided by the borrower
20	purpose	A category provided by the borrower for the loan request
21	title	The loan title provided by the borrower
22	zip_code	The first 3 numbers of the zip code provided by the

Índice	Nombre de la columna	Descripción
		borrower in the loan application
23	addr_state	The state provided by the borrower in the loan application
24	dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income
25	delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
26	earliest_cr_line	The month the borrower's earliest reported credit line was opened
27	inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
28	mths_since_last_delinq	The number of months since the borrower's last delinquency
29	mths_since_last_record	The number of months since the last public record
30	open_acc	The number of open credit lines in the borrower's credit file
31	pub_rec	Number of derogatory public records
32	revol_bal	Total credit revolving balance
33	revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit
34	total_acc	The total number of credit lines currently in the borrower's credit file
35	initial_list_status	The initial listing status of the loan. Possible values are – W, F
36	out_prncp	Remaining outstanding principal for total amount funded
37	out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
38	total_pymnt	Payments received to date for total amount funded
39	total_pymn_inv	Payments received to date for portion of total amount funded by investors
40	total_rec_prncp	Principal received to date
41	total_rec_int	Interest received to date
41	total_rec_int	Interest received to date
42	total_rec_late_fee	Late fees received to date
43	recoveries	post charge off gross recovery
44	collection_recovery_fee	post charge off collection fee
45	last_pymnt_d	Last month payment was received
46	last_pymnt_amnt	Last total payment amount received
47	next_pymnt_d	Next scheduled payment date
48	last_credit_pull_d	The most recent month LC pulled credit for this loan
49	collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections

Índice	Nombre de la columna	Descripción
50	mths_since_last_major_derog	Months since most recent 90-day or worse rating
51	policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
52	application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
53	annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
54	dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income
55	verification_status_joint	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
56	acc_now_delinq	The number of accounts on which the borrower is now delinquent
57	tot_coll_amt	Total collection amounts ever owed
58	tot_cur_bal	Total current balance of all accounts
59	open_acc_6m	Number of open trades in last 6 months
60	open_il_6m	Number of currently active installment trades
61	open_il_12m	Number of installment accounts opened in past 12 months
62	open_il_24m	Number of installment accounts opened in past 24 months
63	mths_since_rcnt_il	Months since most recent installment accounts opened
64	total_bal_il	Total current balance of all installment accounts
65	il_util	Ratio of total current balance to high credit/credit limit on all install acct
66	open_rv_12m	Number of revolving trades opened in past 12 months
67	open_rv_24m	Number of installment accounts opened in past 24 months
68	max_bal_bc	Maximum current balance owed on all revolving accounts
69	all_util	Balance to credit limit on all trades
70	total_rev_hi_lim	Total revolving high credit/credit limit
71	inq_fi	Number of personal finance inquiries
72	total_cu_tl	Number of finance trades
73	inq_last_12m	Number of credit inquiries in past 12 months

A.2. Preprocesado

Los detalles del preprocesado se pueden ver en el propio código, aquí. A resaltar:

- La partición entre train y test se hace manteniendo las proporciones en el target.
- Se ha creado la clase `LoansTransformer`, que puede ejecutar la función fit_transform y transform, típica de los pipelines, de modo que las transformaciones se definan en el fit y se ejecuten con el transform, para que no haya problemas al hacerlo con las particiones de train y test.
- Se han eliminado las columnas con un porcentaje superior al 10 % de nulos.
- Se han eliminado las columnas con una correlación superior al 90 %. Ver Figura 1.
- A la hora de ejecutar los modelos de sklearn necesitaremos hacer previamente un OneHotencoder. La razón para no hacer directamente el OneHotEncoder es que uno de los paquetes de interpretabilidad que usaremos, Lime, necesita tener tanto el modelo con las categorías como el modelo entrenado con una columna por categoría.

El listado de columnas eliminadas y de columnas categóricas se puede consultar aquí, en el valor `self.del_columns`.

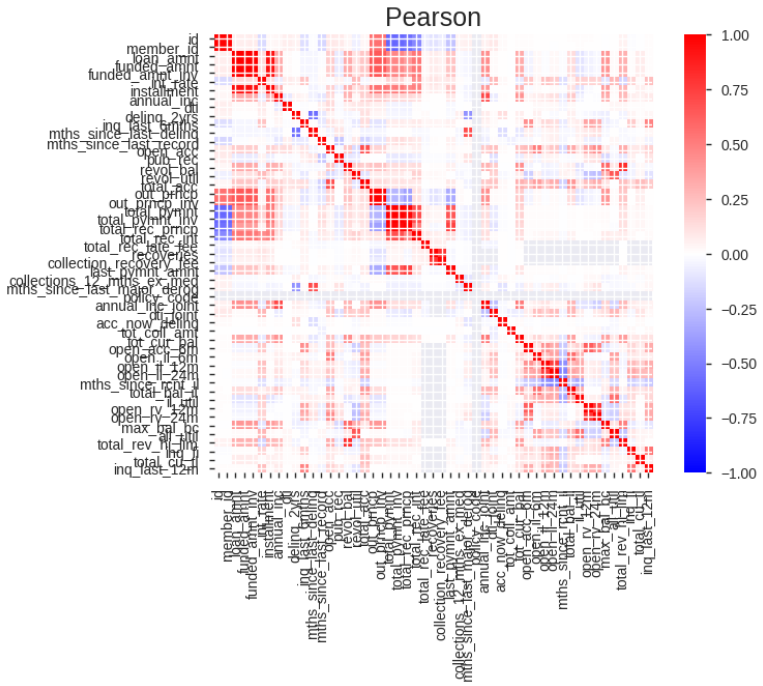

Figura 1: Matriz de correlación entre las distintas variables

B. Modelos

B.1. Hiperparámetros

Método	Parámetros	Espacio de búsqueda
Regresión logística	Penalty	l1,l2
	C	0.001, 0.01, 0.1, 1, 10, 100, 1000
Decision Tree	max_depth	3, 5
	min_samples_split	2, 5
Random Forest	max_depth	10, 15
	min_samples_split	5, 10
	n_estimators	100
Extreme Gradient Boosting	max_depth	2, 4, 6, 8, 10
	min_child_weight	5, 10, 20
	n_estimators	25, 50, 100, 200

Cuadro 9: Espacio de búsqueda de los hiperparámetros

Referencias

[1] Juan Ramón Rallo Julián. El coste del patrón oro, 2018. URL https://bit.ly/2L4uLCP. Publicado en su blog.

[2] Alberto Cano González. El riesgo de crédito en los principales bancos españoles. Publicado en academia.edu, 2019. URL https://bit.ly/2LdxdWf.

[3] Banco Santander. Informe anual, 2018. URL https://bit.ly/2TRPOow.

[4] Alec Zhixiao Lin and Tung-Ying Hsieh. Expanding the use of weight of evidence and information value to continuous dependent variables for variable reduction and scorecard development. *SESUG*, 2014. URL https://www.lexjansen.com/sesug/2014/SD-20.pdf.

[5] Sundar Krishnan. Weight of evidence and information value using python, 2018. URL https://bit.ly/2L60hjG. Publicado en Medium.