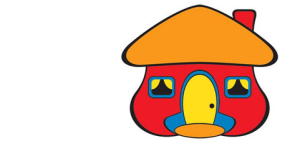
Modelos de clasificación: Random Forest y XGBoost

Natalia Castilla Reyes
astrid.castilla@davivienda.com



Departamento de modelos de parametrización y riesgo de crédito

Diciembre 4, 2020

Resumen

Introducción

Todos los algortimos de ML tienen como punto de partida los siguientes componentes:

- Conjunto de datos: $\wp = (X, y)$, donde X representa una matriz de variables independientes; y es un vector con las variables objetivo.
- Modelo: $f(x; \theta)$, donde $f : x \rightarrow y$
- Función de costo: $\mathbb{C}(y, f(X, \theta))$: permite evaluar el desempeño de nuestro modelo.

Introducción (cont.)

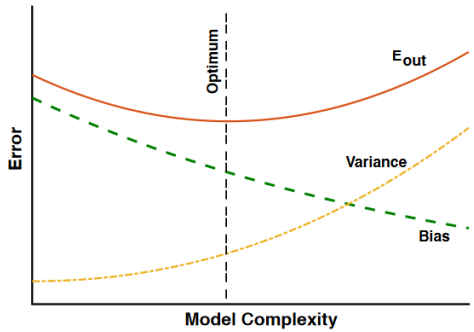
Generalmente, el procedimiento para obtener un resultado es:

- El conjunto de datos se divide en dos: $\wp_{train}$ y $\wp_{test}$.
- El modelo se ajusta minimizando la función de costo usando $\wp_{train}$ $\hat{\theta} = argmin_{\theta}\{\mathbb{C}(y_{train}, f(X_{train}, \theta))\}$.
- Finalmente, el desempeño del modelo se mide calculando la función de costo en $\wp_{test}$.

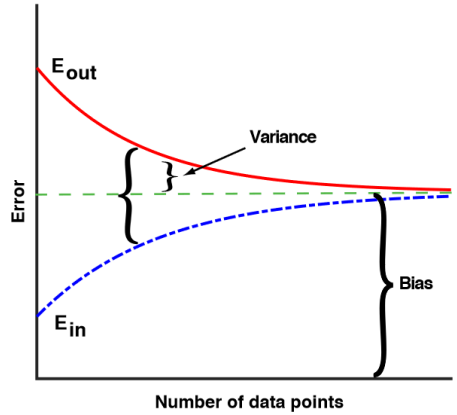
Introducción (cont.)

- Función desconocida: $y = f(x) + \epsilon \rightarrow$ se produce (x_i, y_i), $i = 1, ..., N$. Para describir (x_i, y_i) se tiene un conjunto de hipótesis: $h_i \in \mathbb{H}$.
- El objetivo es seleccionar un $h_i \in \mathbb{H} \longleftrightarrow h_i \approx f$. Adionalmente, se debe evaluar E_{in} y E_{out} para sacar una conclusión 2b.
- Un resultado importante es: $E_{out} = Bias^2 + Var + Noise$. Donde

$$Bias^2 = \sum_i (f(x_i) - \mathbb{E}_D[f(x_i, \hat{D})])^2; \quad Var = \sum_i \mathbb{E}_D[(f(x_i; \theta_D) - \mathbb{E}_D[f(x_i, \hat{\theta}_D)])^2; \quad Noise = \sum_i \epsilon_i^2 \quad (1)$$



(a) Compensación sesgo-varianza en función de la complejidad del modelo.



(b) E_{in} y E_{out} en función del número de datos.

Introducción (cont.)

¿Cómo podemos estimar la generalización de nuestro modelo?

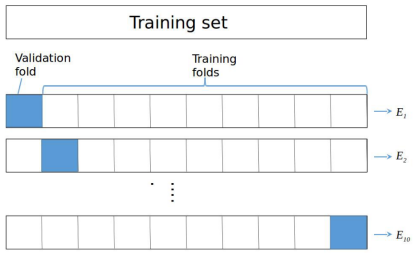


Figura 3: División de los datos de entrenamiento empleando K-Fold CV
$CV_{error} = \frac{E_1+\ldots+E_{10}}{10}$.

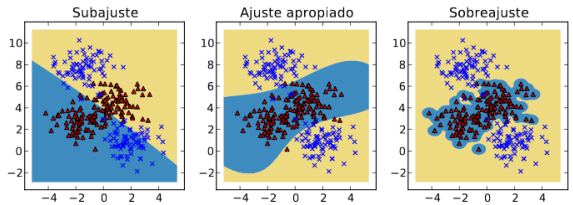


Figura 4: Representación gráfica del problema de varianza y sesgo.

Problema de varianza: $E_{cv}\hat{f} > E_{train}\hat{f}$. Sobreajuste.
Problema de sesgo: $E_{cv}\hat{f} \approx E_{train}\hat{f}$ y $E_{cv}\hat{f} \gg E_{esp}$. Subajuste.

Árbol de decisión:

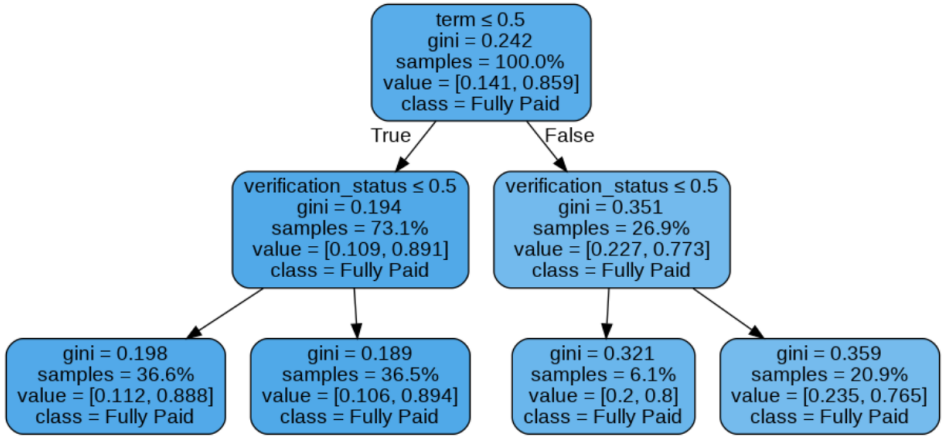


Figura 5: Ejemplo de una árbol de decisión.

El término Gini mide la impureza en cada nodo. Se dice que un nodo es puro cuando gini=0.

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$$

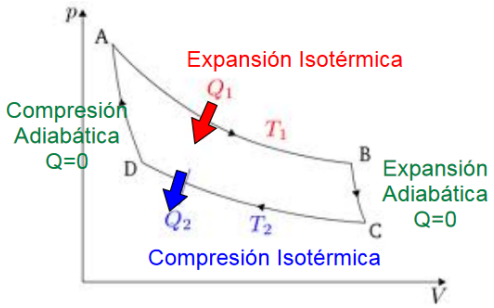
Donde $p_{i,k}^2$ representa la razón entre las variables (value) y la muestra de entrenamiento en cada nodo.

Modelo: Random Forest (cont.)

Sin embargo, podemos ajustar los hiperparamétros del modelo para cambiar la medida de la pureza de cada nodo, entropia:

$$S_i = -\sum_{k=1}^{n} p_{i,k} log(p_{i,k}) \quad donde \quad p_{i,k} \neq 0$$

¿Que es la entropia?



$$E = \frac{3}{2} N k_B T \qquad PV = N k_B T$$

$$Q_1 = \int_A^B P dV = N k_B T_1 \ln \frac{V_B}{V_A}$$

$$Q_2 = -\int_C^D P dV = N k_B T_2 \ln \frac{V_C}{V_D}$$

En la expansión adiabática,

$$dE = -pdV$$

$$\frac{3}{2} N k_B dT = -\frac{N k_B T}{V} dV$$

$$\int_B^C \frac{dV}{V} = -\frac{3}{2} \int_B^C \frac{dT}{T}$$

$$\frac{V_C}{V_B} = \left(\frac{T_1}{T_2}\right)^{3/2} = \frac{V_D}{V_A}$$

$$\frac{Q_2}{T_2} = -\frac{Q_1}{T_1}$$

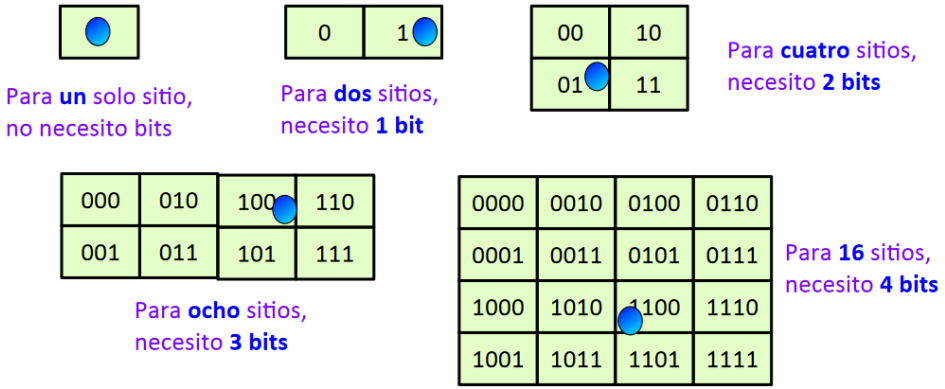
$$\Delta S(A \to B) = \frac{Q_1}{T_1} = N k_B (\ln V_B - \ln V_A)$$

Modelo: Random Forest (cont.)

¿Que es la entropia a nivel microscópico? Entropia: Información promedio que hace falta para saber en qué estado está el sistema.

La información se mide en bits (Shannon)

¿Cuántos bits necesito para identificar dónde está una partícula?



$$\#\text{sitios} = 2^{\#\text{bits}}$$

$$\#\ \text{bits} = \ln_2(\#\ \text{sitios})$$

Si todos los sitios son igualmente probables

$$I_i = \#bits = -\log_2(p), \quad p \equiv 1/\#sitios$$

$$S =< I >= \sum_i p_i I_i = -\sum_i p_i \log_2(p_i)$$

Modelo: Random Forest (cont.)

¿Cuál es la función de costo para problemas de clasificación?

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

Complejidad computacional:

- En cada nodo solo se evalúa una variable. El número de nodos del árbol depende del paramétro de densidad (paramétro de regularización).
- Por lo tanto, la complejidad computacional está dada por: $\mathbb{O}(nxm \log_2(m))$

Modelo: Random Forest (cont.)

Ventajas y desventajas de los árboles de decisión:

- Son sencillos de interpretar.
- Captura relaciones no lineles entre los datos.
- No es necesario estandarizar los datos.
- Las fronteras de decisión son ortogonales.
- Es muy sensible a pequeñas variaciones en los datos.
- Si no regularizamos el CART se puden tener problemas de varianza.

Para dar solución al problema de varianza del DT se introdude el concepto de ensamble. Para construir el esamble se debe tener en cuenta: Reducir las correlaciones entre los modelos que componen el esamble. Esto asegura que la varianza se reduzca.

Modelo: Random Forest (cont.)

Método de esamble:

Podemos demostrar que las expresiones de la ecuación (1) mantinen su forma, sin embargo, $\hat{f}(x;\theta)$ se debe cambiar por su versión generalizada:

$$\hat{g}_{\mathbb{L}}^{\mathbb{A}} = \frac{1}{M}\sum_{m=1}^{M}\hat{g}_{\mathbb{L}}(x;\theta_m)$$

Se puede observar que $\hat{g}_{\mathbb{L}}^{\mathbb{A}}$ es una suma de estimadores, su varianza depende de manera indirecta de las correlaciones entre los estimadores individuales del conjunto.

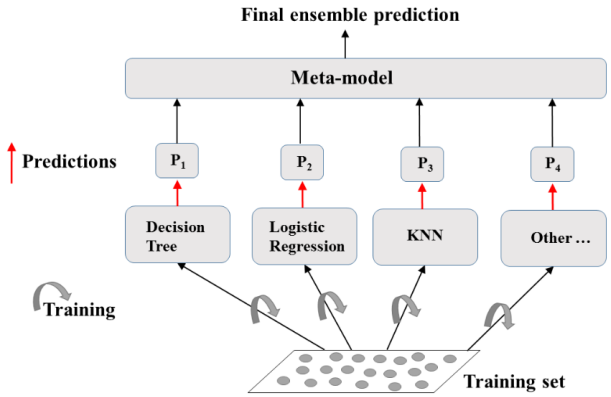


Figura 6: Explicación visual del método de ensamble.

Modelo: Random Forest (cont.)

Teniendo en cuenta las siguientes definiciones:

$$\mathbb{E}_{\mathbb{L},\theta_m}[\hat{g}_{\mathbb{L}}(x,\theta_m)] = \mu_{\mathbb{L},\theta_m}(x); \quad \mathbb{E}_{\mathbb{L},\theta_m}[\hat{g}_{\mathbb{L}}(x,\theta_m)^2] - \mathbb{E}_{\mathbb{L},\theta_m}[\hat{g}_{\mathbb{L}}(x,\theta_m)]^2 = \sigma^2_{\mathbb{L},\theta_m}(x)$$

$$\mathbb{E}_{\mathbb{L},\theta_m}[\hat{g}_{\mathbb{L}}(x,\theta^*_m)\hat{g}_{\mathbb{L}}(x,\theta_m)^2] - \mathbb{E}_{\mathbb{L},\theta_m}[\hat{g}_{\mathbb{L}}(x,\theta^*_m)\hat{g}_{\mathbb{L}}(x,\theta_m)]^2 = C_{\mathbb{L},\theta,\theta^*}$$

$$\rho(x) = \frac{C_{\mathbb{L},\theta,\theta^*}}{\sigma^2_{\mathbb{L},\theta_m}(x)}$$

Las definiciones de la ecuación (1) pueden escribirse como:

$$Var(x) = (\rho(x) + \frac{1-\rho(x)}{M})\sigma^2_{\mathbb{L},\theta}; \quad Bias^2(x) = (f(x) - \mu_{\mathbb{L},\theta})^2$$

Note que si $M \to \infty$ (esambles muy grandes), se reduce significativamente la varianza. Para ensambles completamente aleatorios estos no están correlacionados es decir: $\rho(x) = 0$. Para un conjunto aleatorio siempre se pueden añadir más modelos sin aumentar el sesgo. Esta observación se encuentra detrás del inmenso poder del método de Random Forest.

Modelo: Random Forest

- El estimador básico es el árbol de decisión.
- Reduce la variaza del DT.
- Cada estimador es entrenado en una muestra seleccionada mediante la técnica bootstrapping.

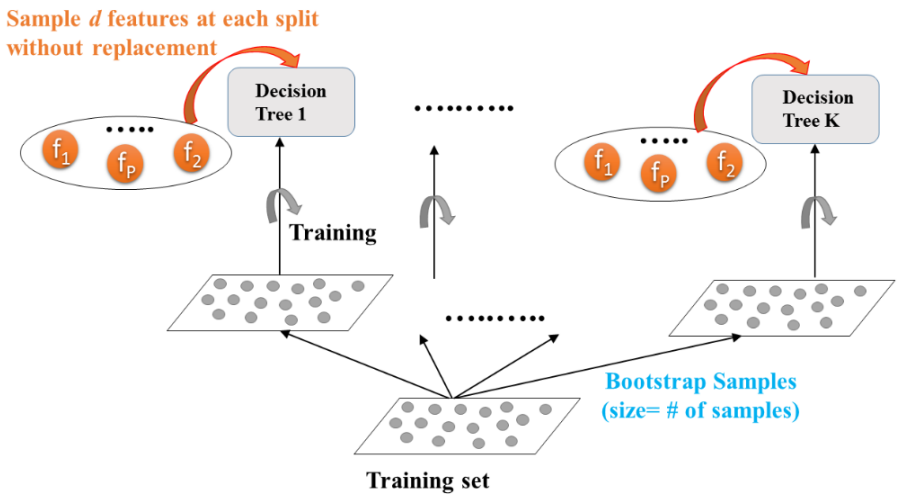


Figura 7: Esquema de entrenamiento del algoritmo de Random Forest.

Modelo: Random Forest

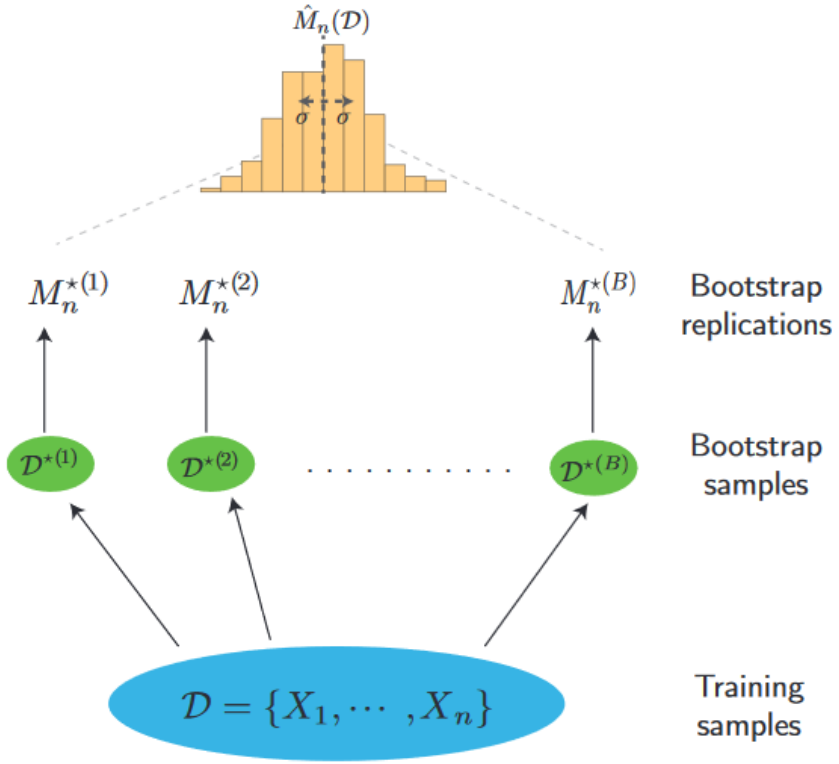


Figura 8: Representación visual de la técnica de la bootstrapping. $M_n^{*(B)}$ es la mediana de la cantidad de interés en una muestra B. En cada una de las muestras $D^{*(B)}$ se incluyen elementos repetidos.

Modelo: XGBoost

- Se compone por $\hat{y}$ "débiles"(CARTs). Cada modelo se entrena secuencialmente, y cada resultado se usa para realizar correcciones en $\hat{y}_{t+1}$ posterior.
- Cada $\hat{y}_t$ es entrenado usando los errores residuales de su predesor.
- En cada iteración los errores se reducen utilizando el método de GD (Gradient Descent)
- Cada DT $(g_j(x))$ se parametriza:
  - q(x), $q : x \in \mathbb{R}^d \rightarrow \{1, 2, 3.., T\}$ donde T es el número de hojas de $g_j(x)$.
  - $w \in \mathbb{R}^T$ valor de predicción para cada hoja de $g_j(x)$, $w_q(x_i) = g_j(x_i)$.

Función de costo:

$$\mathbb{C}(X, g_A) = \sum_{i=1}^{N} \mathbb{L}(y_i, \hat{y}_i) + \sum_{j=1}^{M} \Omega(g_j)$$

Modelo: XGBoost (cont.)

La función de regularización $\Omega(g)$ par XGBoost es:

$$\Omega(g) = \gamma T + \frac{1}{2}||w||^2$$

Debido a que es un método de esamble iterativo, se define la familia $\hat{y}_i^{(t)}$:

$$\hat{y}_i^{(t)} = \sum_{j=1}^{t} g_t(x_i) = \hat{y}_i^{(t-1)} + g_t(x_i)$$

Para $t \to \infty$ se puede realizar una expansión de Taylor en $\mathbb{C}$.

$$\mathbb{C} = \sum_{i=1}^{N} \mathbb{L}(y_i, \hat{y}_i^{t-1} + g_t(x_i)) + \Omega(g_t) \approx \mathbb{C}_{t-1} + \Delta\mathbb{C}_t$$

En donde:

$$\Delta\mathbb{C} = a_i g_t(x_i) + \frac{1}{2} b_i g_t(x_i)^2 + \Omega(g_t); \quad a_i = \partial_{\hat{y}_i^{t-1}} \mathbb{L}(y_i, \hat{y}_i^{t-1}) \quad b_i = \partial^2_{\hat{y}_i^{t-1}} \mathbb{L}(y_i, \hat{y}_i^{t-1})$$

Modelo: XGBoost (cont.)

Se pueden encontrar los paramétros adecuados para miminizar $\Delta\mathbb{C}$.
Teniendo en cuenta:

$$j : I_j = \{i : q_t(x_i) = j\},$$

Donde j corresponde a una hoja del DT.

$$\Delta\mathbb{C}_t = \sum_{j=1}^{T}[A_j w_j + \frac{1}{2}(B_j + \lambda)w_j^2] + \gamma T$$

Tomando el gradiente de la anterior expresión:

$$w_j^{opt} = -\frac{A_j}{B_j + \lambda}; \quad \Delta\mathbb{C}^{opt} = -\frac{1}{2}\sum_{j=1}^{T}\frac{A_j^2}{B_j + \lambda} + \gamma T$$

Modelo: XGBoost (cont.)

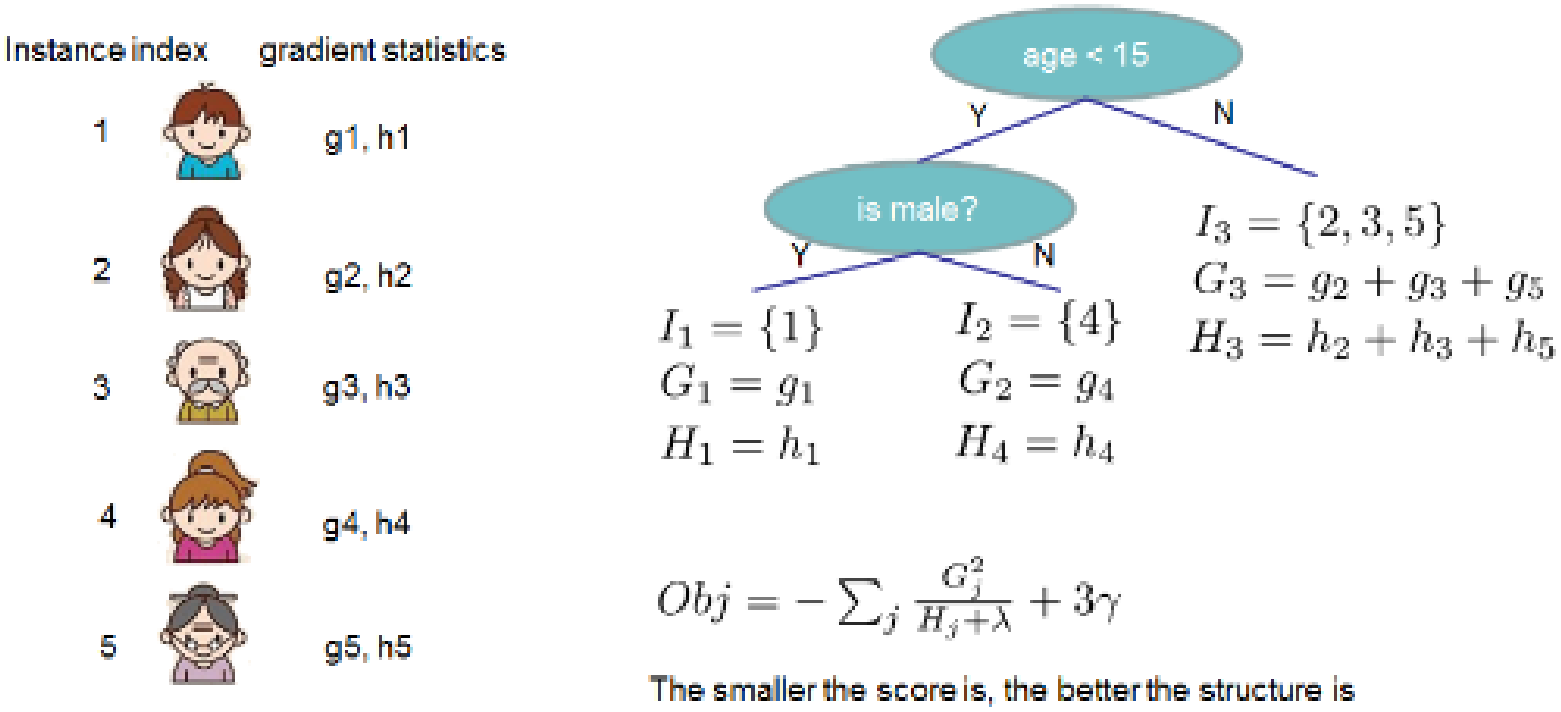


Figura 9: Definición esquemática del algoritmo.

Descripción del problema

Sobre el conjunto de datos: Contiene la información de todos los préstamos emitidos entre el periodo 2007-2015, incluida la situación actual de los préstamos (corriente, atrasado, pagado totalmente, etc.) y la información más reciente sobre los pagos. Entre otras caracteristicas, se incluyen los puntajes de crédito, el número de consultas financieras, la dirección, incluidos los códigos postales, y el estado y los cobros, entre otros. El conjunto de datos es una matriz de unas 890 mil observaciones y 75 variables. Se proporciona un diccionario de datos en un archivo separado.

Próposito del análisis: Se contruirán dos algoritmos ML (RF y XGBoost) para la predicción de los morosos basado en ciertas variables presentes en el conjunto de datos. El objetivo principal es identificar correctamente los impagos (Verdaderos positivos) para que el club de préstamos pueda decidir si una persona es apta para sancionar un préstamo o no en el futuro.

EDA

Para aplicar todos los conceptos anteriormente estudiados se seleccionó un conjunto de datos relacionado con el tema financiero. El estudio y prepocesamiento de los datos siguieron los siquientes pasos:

- Exploración superficial de los datos a través de *pandas* − *profiling*.
- Análisis de distribuciones de algunas variables de interés.
- Adecuación del marco de datos para realizar un pequeño análisis demográfico.
- Preprocesamiento de los datos: imputación de valores nulos, balanceo de los datos, estudio de correlación, estudio de importancia de las variables para el modelo según su contenido y detección de valores anómalos.

A continuación se muestran los resultados obtenidos de esta fase:

EDA (cont.)

- Muchos de los préstamos solicitados están en el rango de 10,000 USD a 20,0000 USD.
- La cantidad de dinero solicitado va aumentando cada año.

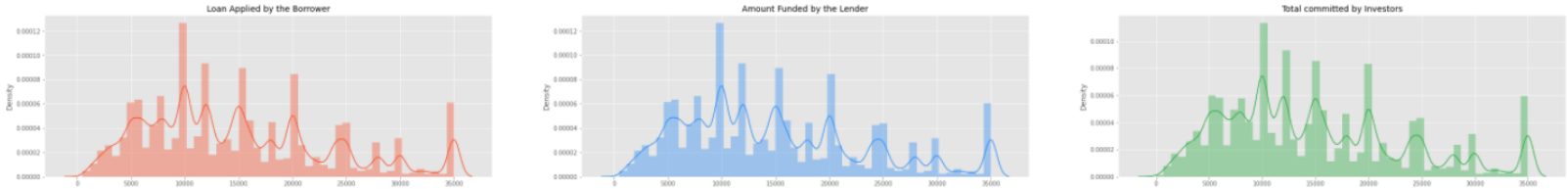


Figura 10: Distribuciones para las variables:loan amount, funded amount e investor funds. (Las dos últimas corresponden a la cantidad de endeudamiento de cada uno).

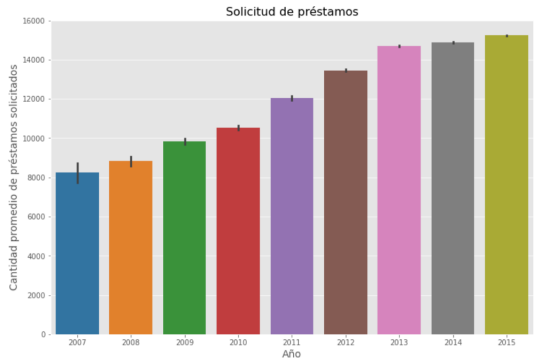
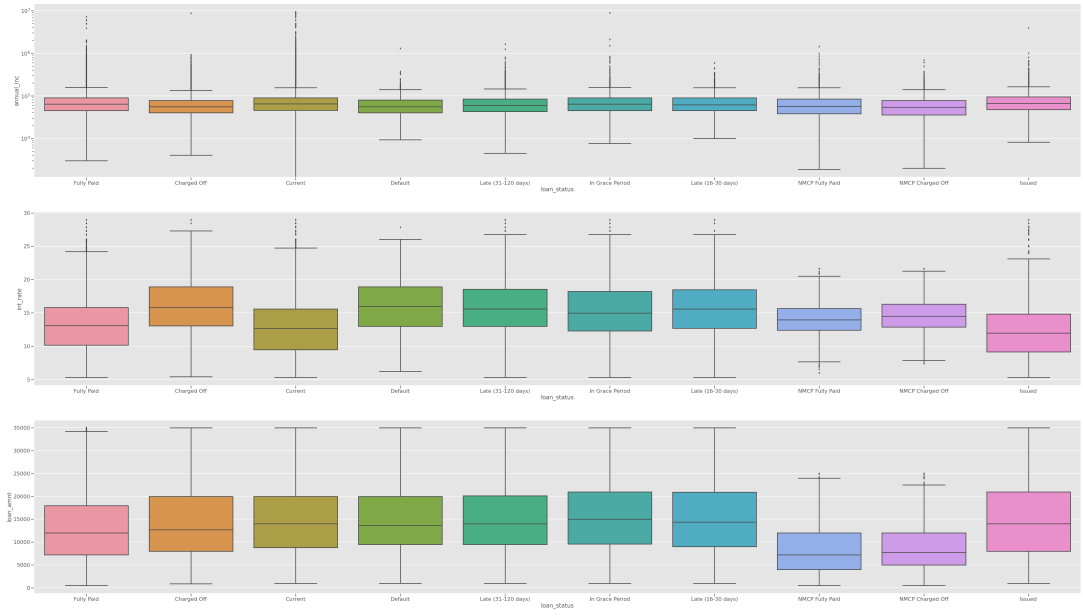


Figura 11: Cantidad promedio de préstamos solicitados por año.

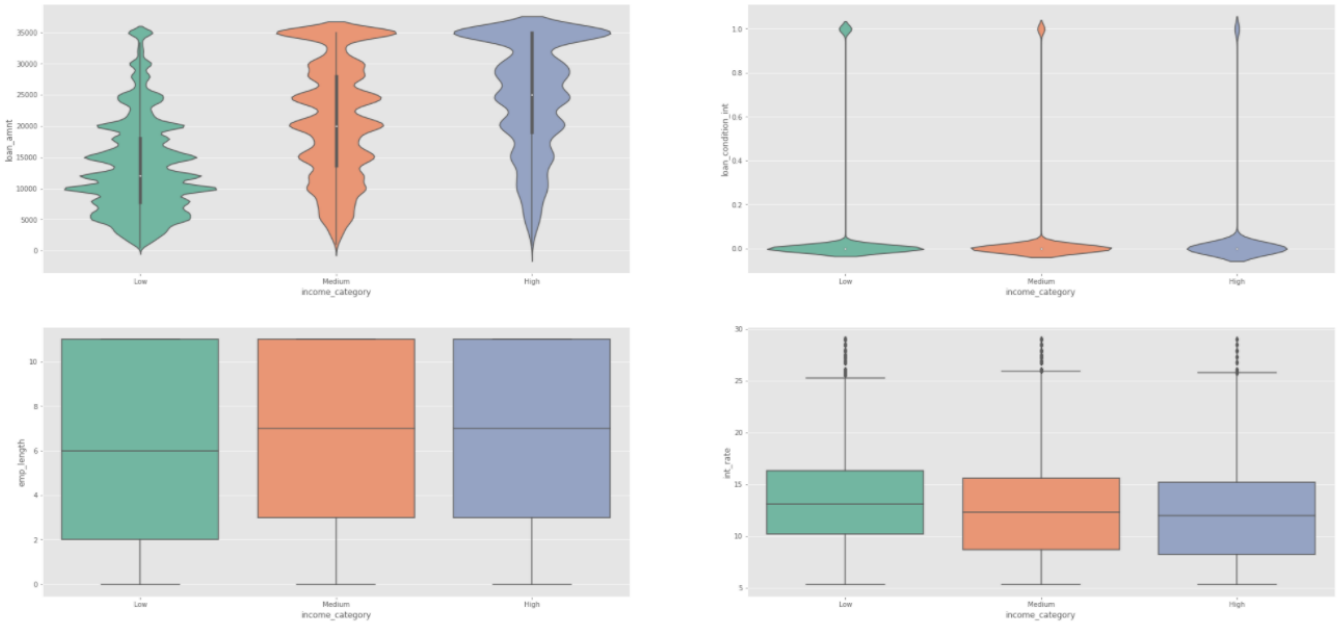
EDA (cont.)

- Vemos que las tasas de interés son más altas para aquellos créditos que tienen riesgo de ser un mal préstamo.
- Las personas que NCMCP tien la cantidad de más baja de dinero solicitado.



EDA (cont.)

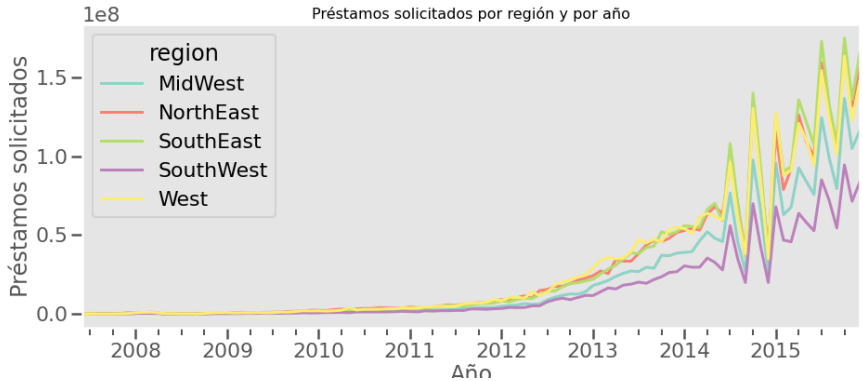
- Bajos ingresos: Prestatarios que tienen un ingreso anual menor o igual a 100.000 usd.
- Ingresos medios: Prestatarios que tienen un ingreso anual superior a 100.000 usd pero inferior o igual a 200.000 usd.
- Ingresos altos: Prestatarios que tienen un ingreso anual superior a los 200.000 dólares.

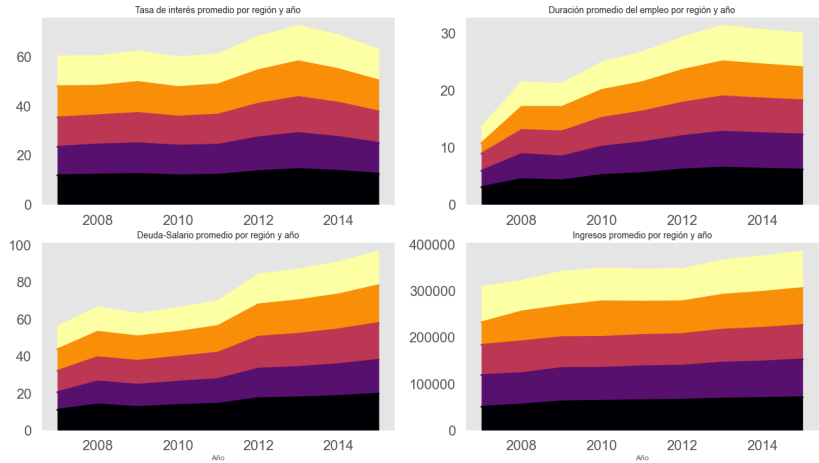


EDA (cont.)

Análisis demográfico:

- Las regiones SouthEast, West y NorthEast tienen la mayor cantidad de préstamos solicitados.
- West, SouthWest tienen un rápido crecimiento en la variable debt-to-income desde el 2012.
- West y SouthWest tienen un decrecimiento en las tasas de interés (Esto puede explicar quizá el incremento en la variable deuda-ingreso).





EDA (cont.)

Distribución operacional y análisis de riesgo por estado:

Figura 12: Análisis de riesgo por estado.

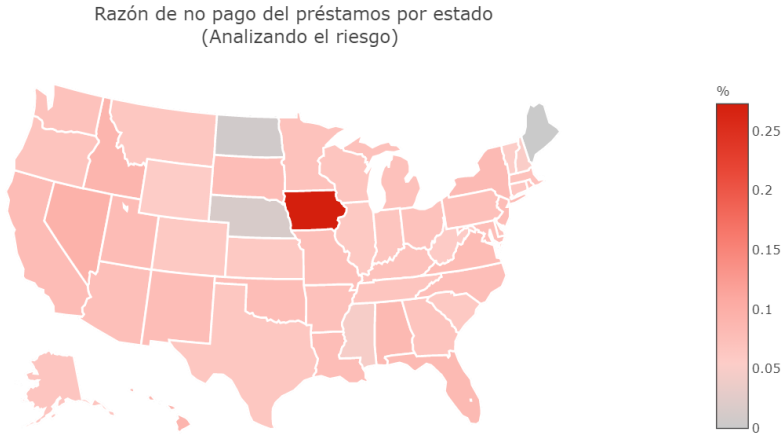
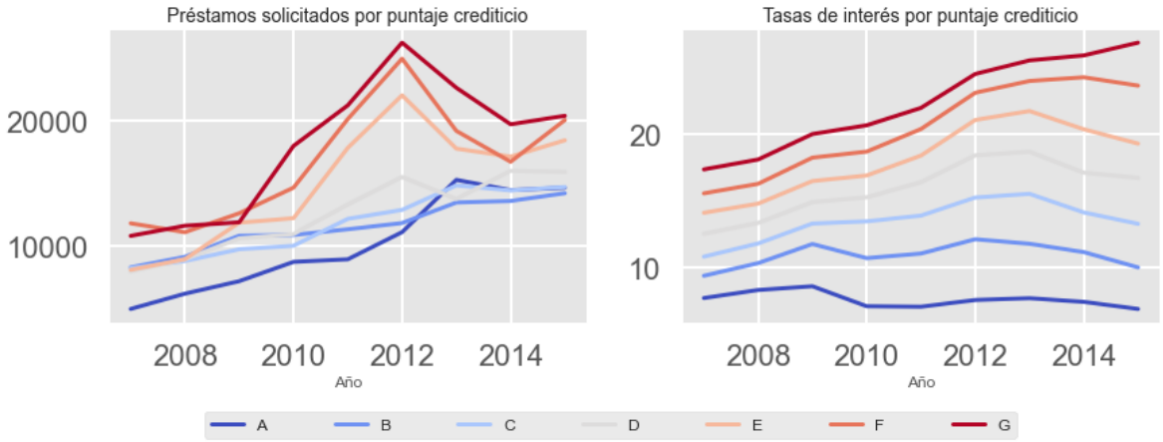


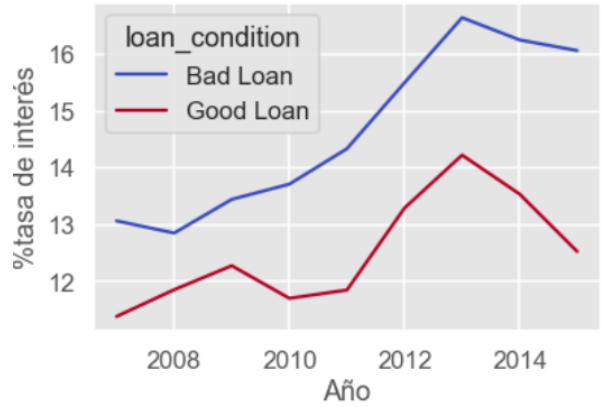
Figura 13: Préstamos solicitados por estado

- Para realizar la Figura 13 se utilizaron las variables: cantidad de préstamos solicitados, ingresos anuales, tasa de intéres.
- Para la figura 12 se utilizaron las variables: cantidad de créditos no pagados, cantidad de malos préstamos, duración del empleo, dit (relación calculada usando los pagos mensuales totales de la deuda sobre el total de las obligaciones de la deuda.)

EDA (cont.)

- Resultados para la Figura 13: California, Texas, Nueva York y Florida son los estados en los que se emitieron más préstamos. California, Texas y Nueva York están todos por encima del promedio de ingresos anuales (con la exclusión de Florida), esto puede dar una posible indicación de por qué la mayoria de los préstamos se emiten en estos estados.
- Resultados para la Figura 12: California y Texas parecen tener el menor riesgo y el mayor rendimiento posible para los inversores.





Resultados de los modelos: Random Forest

- Para tener un mejor desempeño del algoritmo se balancearon los datos $n_{samples}/(n_{classes} * np.bincount(y))$.
- Se realizó el ajuste de los siguientes hiperparamétros: maxdepth=2, maxfeatures='sqrt', minsamplesleaf=0.18, nestimators=200

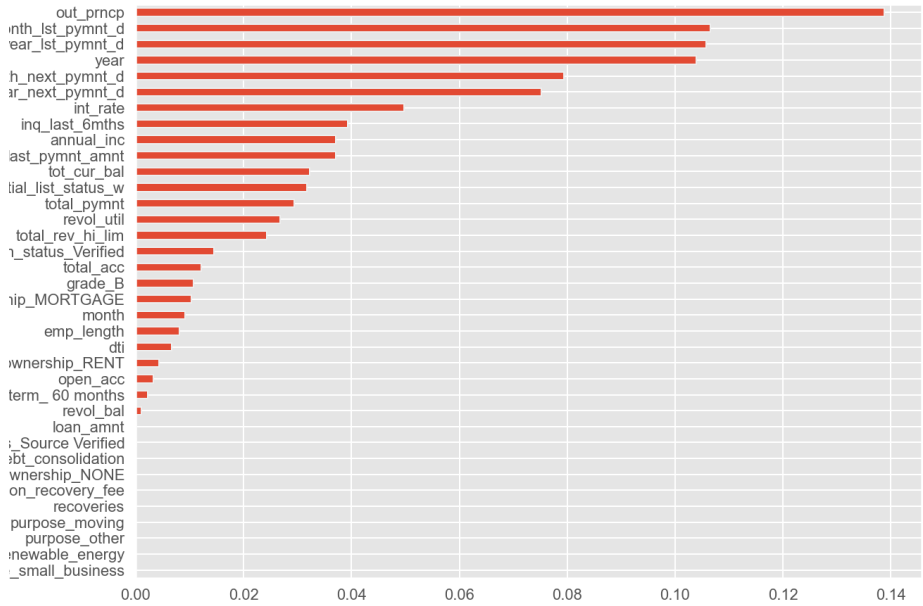


Figura 14: Importancia de las variables en la toma de decisión del algortimo.

Resultados de los modelos: Random Forest (cont.)

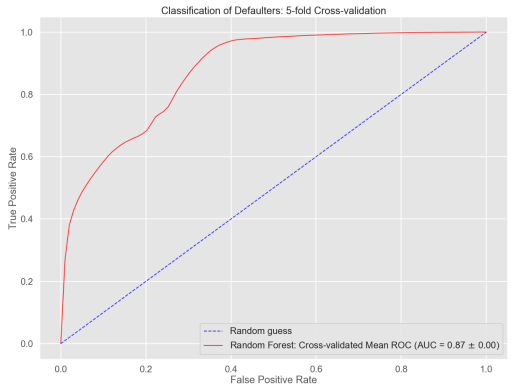


Figura 15: Evaluación del modelo en los datos $\mathbb{D}_{train}$ mediante C-V.

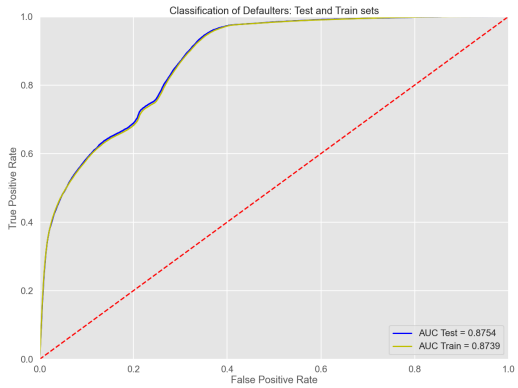


Figura 16: Evaluación del modelo en los datos $\mathbb{D}_{test}$ y $\mathbb{D}_{train}$.

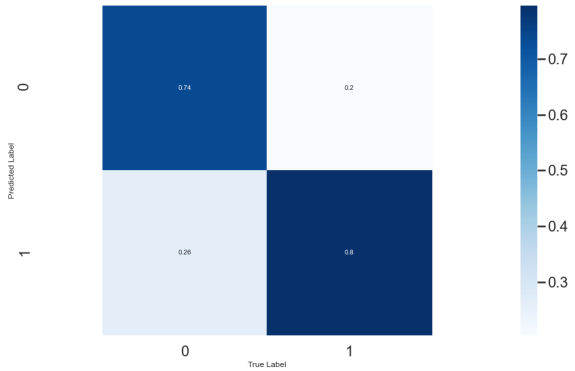


Figura 17: Matriz de confusión.

Resultados de los modelos: XGBoost

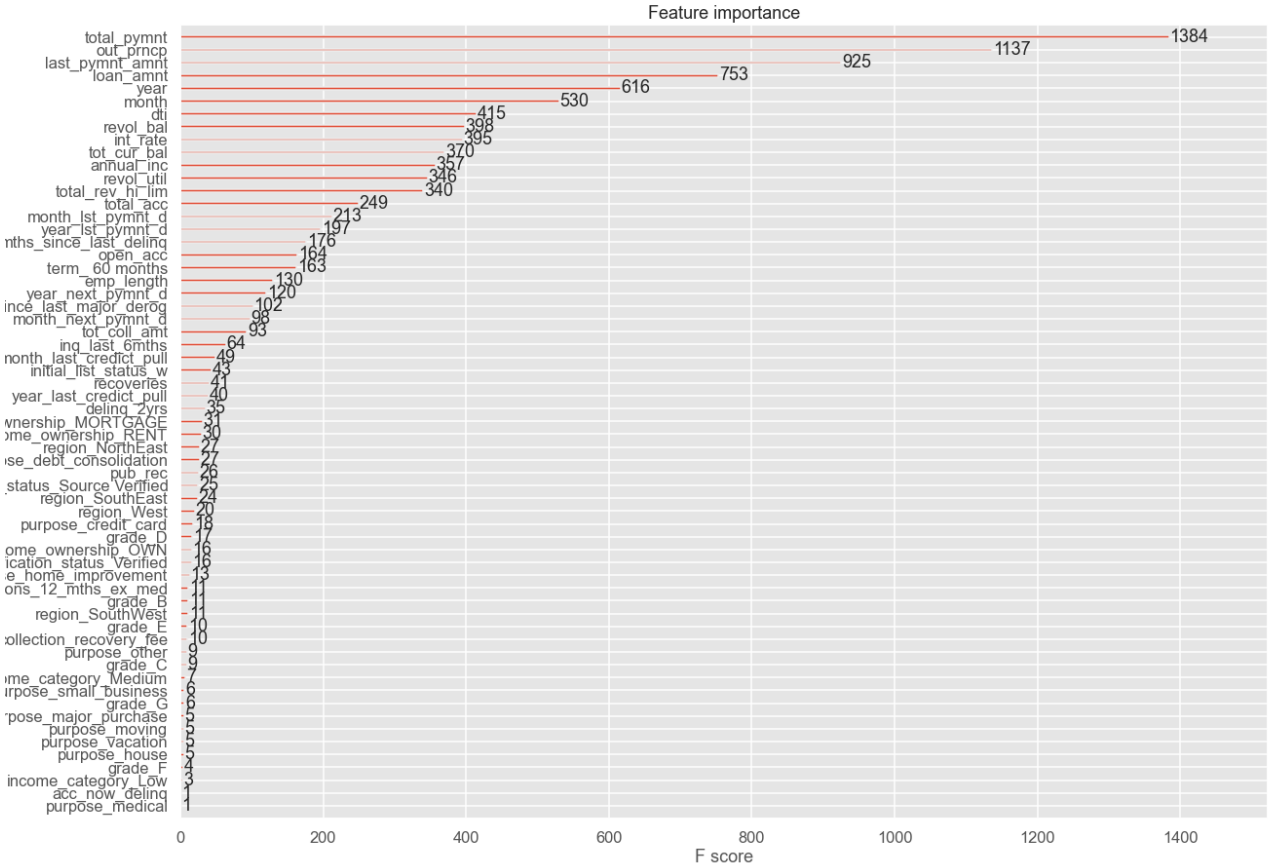


Figura 18: Importancia de las variables en la toma de decisión del algoritmo.

Resultados de los modelos: XGBoost (cont.)

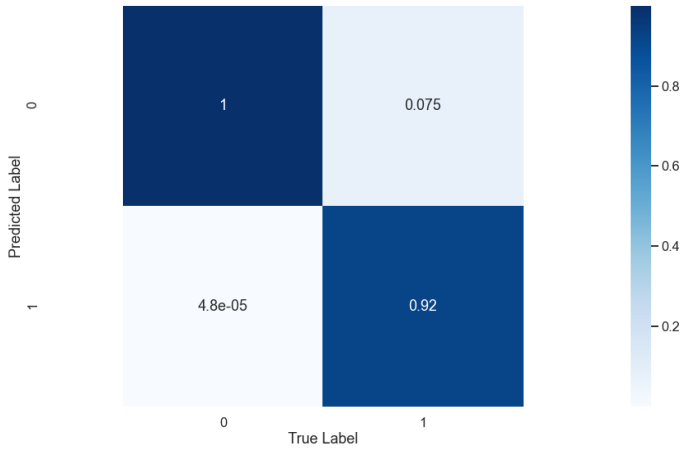


Figura 19: Matriz de confusión para XGboost.

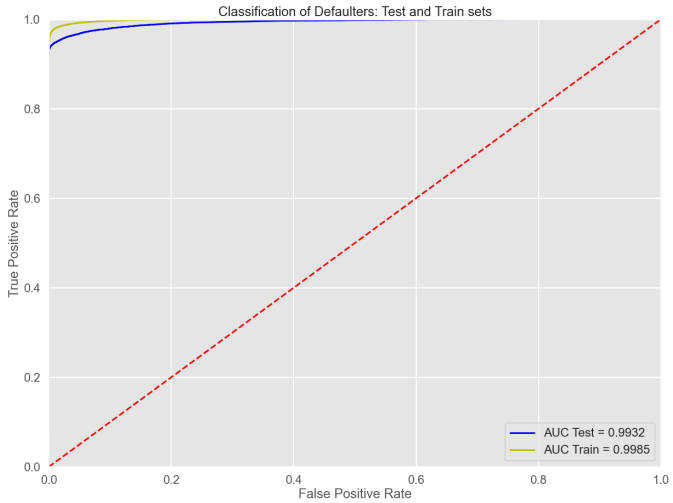


Figura 20: Evaluación del modelo en los datos $\mathbb{D}_{test}$ y $\mathbb{D}_{train}$.

LIME:

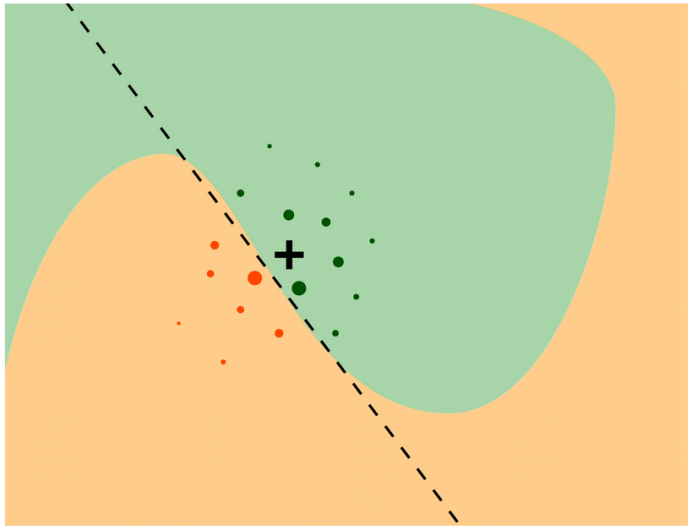


Figura 21: Las áreas coloreadas corresponden a regiones de decisión para un modelo de clasificación binario complejo. La cruz negra indica la observación de interés. Los puntos corresponden a datos artificiales. La linea de puntos representa un modelo lineal simple ajustado a los datos artificiales. El modelo simple explica el comportamiento local del modelo de la caja negra alrededor de una vecindad.

Interpretación de los modelos mediante: LIME y SHAP (cont.)

Notación:

- f modelo de caja negra, XGBoost, Random Forest...
- x^* observación de interés.
- $g_i \in \mathbb{G}$, $\mathbb{G}$ conjunto de modelos sencillos de fácil interpretación.
- Función de costo $\mathbb{L}$.
- $\nu(x^*)$ vecindario de interés.
- $\Omega(g)$ función de regularización de cada función g

$$\hat{g} = argmin_{g \in \mathbb{G}} L\{f, g, \nu(x^*)\} + \Omega(g)$$

Interpretación de los modelos mediante: LIME y SHAP (cont.)

- Los operadores $f()$ y $g()$ se aplican en dos espacios vectoriales diferentes. Para el modelo de cada negra: $f(x) : \mathbb{X} \rightarrow \mathbb{R}$, en donde $\mathbb{X}^p \quad p \equiv dimensión$. Para el modelo sencillo se tiene: $g(x) : \tilde{\mathbb{X}} \rightarrow \mathbb{R}$, en donde $X^q \quad q \ll p$

Algoritmo:

1. Permutar los datos $\mathbb{D}^*$.
2. Calcular la distancia entre los datos permutados y los datos originales*.
3. Hacer las predicciones en los nuevo datos usando el modelo $f()$.
4. Escoger las variables que mejor describen los resultados del modelo a partir del $\mathbb{D}^*$.
5. Ajustar un modelo sencillo con los datos $\mathbb{D}$. Con las variables escogidas en el punto anterior y con los resultados de similaridad como pesos.

Resultados para Random Forest y XGBoost:

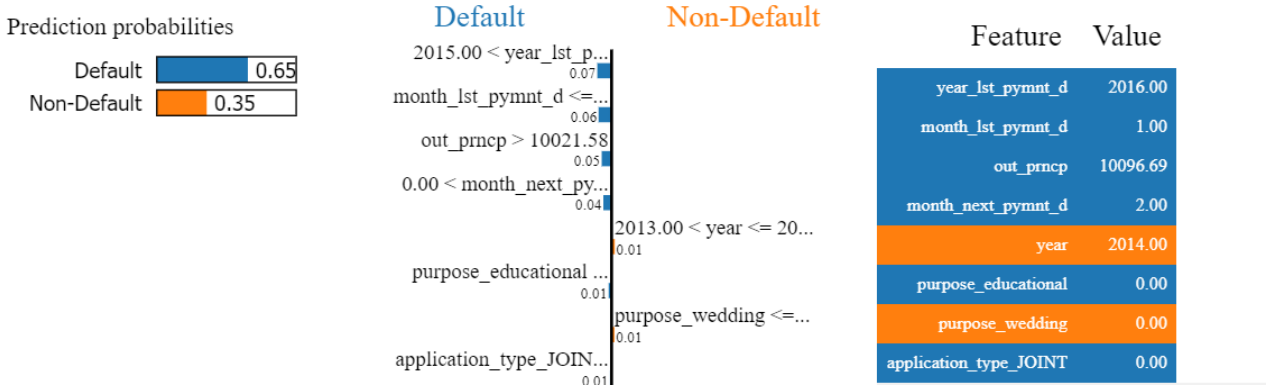


Figura 22: Resultados de LIME para RF y el cliente 2020.

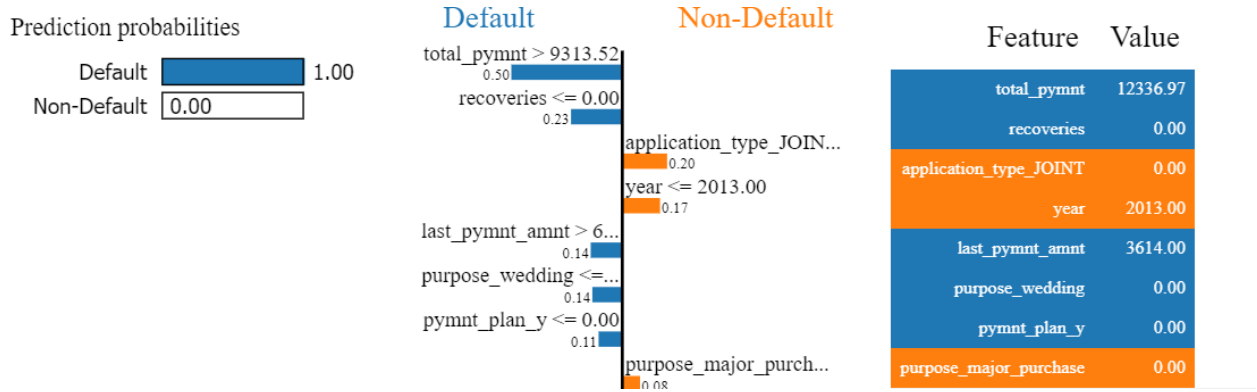


Figura 23: Resultados de LIME para XGBoost y el cliente 2020.

Shapley Additive Explanations

- El algoritmo toma las ideas introducidas por Lloyd Shapley para el desarrollo teórico de los juegos cooperativos.
- En este caso las variables son los jugadores, $f()$ define la cooperación entre ellos. En este caso la ganacia del juego es la predicción del modelo $\hat{y}$.

El valor de Shapley para un punto x^* es:

$$\phi(x^*, j) = \frac{1}{p!} \sum_{J} \Delta^{j|\pi(J,j)}(x^*)$$

En donde $\delta^{j|\pi(J,j)}$ es:

$$\Delta^{L|J}(x^*) \equiv E_{\bar{X}}\{f(\bar{X})|X^{l_1} = x^{*l_1}, ..., X^{l_M} = x^{*l_M}, X^{j_1} = x^{*j_1}, ..., X^{j_K} = x^{*j_K}\}$$

$$-E_{\bar{X}}\{f(\bar{X})|X^{j_1} = x^{*j_1}, ..., X^{j_K} = x^{*j_K}\}$$

Shapley Additive Explanations (cont.)

En otras palabras, $\phi(x^*, j)$ es la media de la importancia de cada variable teniendo en cuenta todas las permutaciones posibles. Algunas propiedades:

- Simetria: si las variables j y k son intercambiables para cualquier conjunto $S \subseteq \{1, 2, ..., p\}$
  - $\Delta^{j|S}(x^*) = \Delta^{k|S}(x^*) \rightarrow \phi(x^*, j) = \phi(x^*, k)$
- Si j no contribuye a $\hat{y}$ en algún conjunto S:
  - $\Delta^{j|S}(x^*) = 0 \rightarrow \phi(x^*, j) = 0$
- Podemos realizar transformaciones lineales.
- Localidad:
  - $f(x^*) - E\{f(\bar{X})\} = \sum_{j=1}^{p} \phi(x^*, j)$

Shapley Additive Explanations (cont.)

Resultados para los modelos de Random Forest y XGBoost (resultados locales):

Figura 24: Resultados (Shapley-Values) para RF, utilizando el algoritmo: TreeExplainer. Para un registro particular.

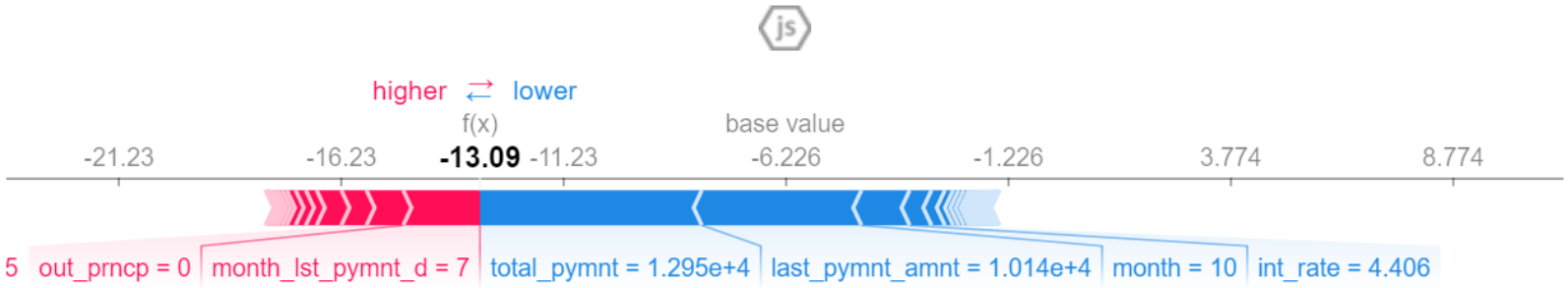


Figura 25: Resultados (Shapley-Values) para Xgboost, utilizando el algoritmo: TreeExplainer. Para un registro particular.

Shapley Additive Explanations (cont.)

Resultados para los modelos de Random Forest y XGBoost (resultados global, promedio de los Shapley Values):

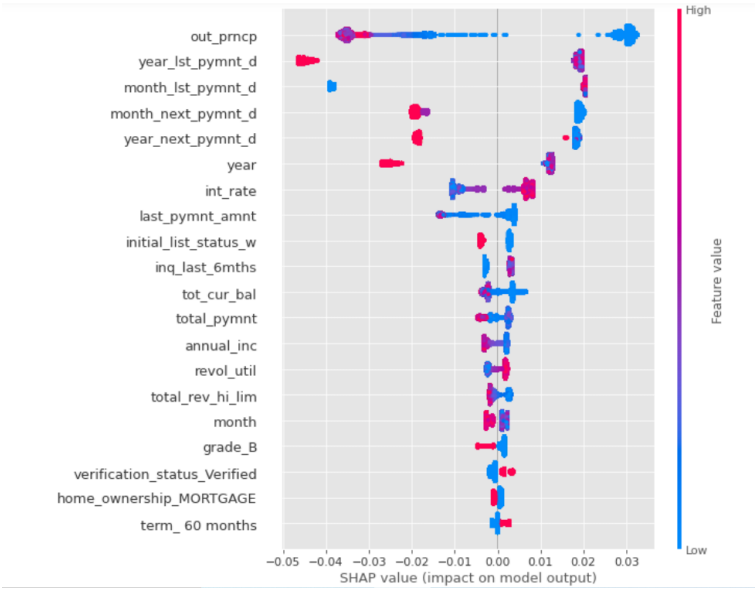


Figura 26: Resultados de la importancia de las variables según SHAP para el modelo de RF

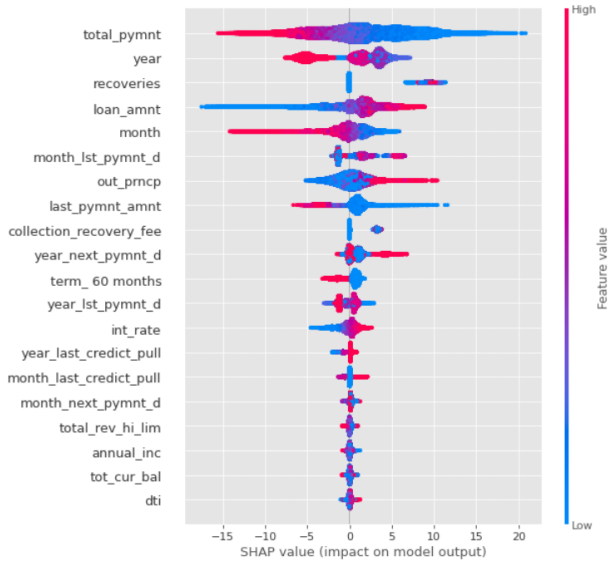


Figura 27: Resultados de la importancia de las variables según SHAP para el modelo de XGBoost.

Referencias

[1] Pankaj Mehta, Ching-Hao Wang. *A high-bias, low-variance introduction to Machine Learning for physicists*. https://arxiv.org/pdf/1803.08823.pdf. physics.comp-ph, arXiv, 2019.

[2] Tomasz Burzykwosky, *Explanatory Model Analysis*. https://pbiecek.github.io/ema/preface.html. CRC Press, 2020.

[3] Edden M. Gerber, *A new perspective on Shapley values, part I: Intro to Shapley and SHAP*. https://edden-gerber.github.io/shapley-part-1/.Noviembre, 2019.

[4] Priyanshu Sharma, *Lending Clud DATA*. https://www.kaggle.com/braindeadcoder/lending-club-data. Diciembre, 2018.