Score crediticio con Regresión Logística y Máquinas de Soporte Vectorial

Cervantes Artavia, Joshua. Monge Cordonero, Moisés. Sabater Guzmán, Daniel.[1]

joshua.cervantes@ucr.ac.cr, moises.mongecordonero@ucr.ac.cr, daniel.sabater@ucr.ac.cr

Resumen

En la actualidad dada la capacidad computacional con la que se cuenta se ha comenzado a implementar algoritmos de machine learning en distintas áreas. Entre ellas el manejo de riesgo crediticio. No obstante, los algoritmos de machine learning suelen ser complejos y debido a la falta de datos no se suele encontrar una basta cantidad de artículos que expliquen de forma detallada el cómo es posible construir un modelo de score crediticio empleando estos algoritmos. En este trabajo se construyen dos modelos de score crediticio haciendo uso de los algoritmos de machine learning, regresión logística (ampliamente utilizado) y máquinas de soporte vectorial. Para ello se empleó una base de datos que corresponde a clientes que contaban con un crédito en un banco en el sur de Alemania en el periodo comprendido entre 1973 y 1975, dicha base muestra cuando un cliente ha incurrido o no en un incumplimiento del contrato crediticio. Para ajustar los modelos se han escalado variables, se ha empleado validación cruzada y se han medido los resultados con la curva ROC y el AUC. Al aplicarse los modelos se ha encontrado la importancia en la manipulación de los datos. Tanto el modelo de máquinas de soporte vectorial como regresión logística obtuvieron resultados de AUC cercanos al 0.80. No obstante, el que obtuvo un resultado ligeramente más alto fue la máquina de soporte vectorial. Se concluye con la importancia que puede tener la base de datos para el ajuste o no del algoritmo. Además, a pesar de que el algoritmo de máquinas de soporte vectorial muestra un mejor grado de ajuste comparado con regresión logística, este primero representa un mayor costo computacional.

Palabras clave

Machine Learning, aprendizaje supervisado, riesgo crediticio.

Introducción

El manejo de riesgo de crédito es una tarea de gran relevancia para las entidades financieras, por lo que estas han desarrollado métodos que le permiten disminuir estos riesgos, tratando de predecir aquellas personas que no pagarán su crédito bajo las condiciones establecidas. A su vez con el aumento de la capacidad computacional, se han comenzado a implementar cada vez más algoritmos de Machine Learning (ML) en distintas áreas, entre ellas el de manejo de riesgos de crédito.

Al adentrarse al tema hay que tener presente la teoría detrás de la gestión de riesgos de crédito. Desde un punto de vista jurídico, un crédito es toda operación formalizada por un intermediario financiero, que bajo algún riesgo provee fondos a otro individuo que es el deudor (Banco Central de Costa Rica 1998). Un riesgo crediticio es una posibilidad de pérdida económica debido al incumplimiento de las condiciones pactadas (CONASSIF, 2017). La existencia de estos riesgos de incumplimiento o default, dan origen a técnicas y metodologías para determinar, cuantificar y prevenir transacciones que aumenten este riesgo.

[1] Estudiantes Escuela de Matemática, Universidad de Costa Rica.

De querer manejar el riesgo crediticio, es que surge el credit scoring que como lo define Gutiérrez (2007), se constituyen en algoritmos los cuales evalúan de manera automática, el riesgo de crédito de un solicitante de financiamiento; por lo que score crediticio es la práctica realizada antes de iniciar un crédito por parte de las entidades financieras para determinar la probabilidad de impago de la persona.

De acuerdo con lo expuesto por Ławrynowicz y Tresp (2014), Wehle (2017), Shalev-Schwartz y Ben-David (2014), se puede definir ML como el uso de distintos algoritmos o métodos que tienen como objetivo desarrollar programas computacionales, que son capaces de encontrar patrones en conjuntos de datos y de cierta forma aprender de estos.

Entre los algoritmos de ML implementados para score crediticio están el de máquinas de soporte vectorial (SVM), regresión logística (LR), random forest, redes neuronales y otros algoritmos que permiten la clasificación binaria y son empleados para predicción. Según Hältuf (2014) usando como medida de ajuste la curva característica operativa del receptor (ROC) y área bajo esta (AUC), SVM (computacionalmente más costoso) supera ligeramente a LR que es implementado por distintas instituciones financieras. Chow (2007), realiza una comparativa de desempeño entre distintos modelos mediante la curva ROC y AUC, exhaustividad, precisión, exactitud y F1-score, con el modelo SVM obtiene buenos resultados. Osisanwo y col. (2017), destacan SVM como uno de los métodos más novedosos de clasificación y predicción, sin embargo, no todos los modelos se ajustan de la misma manera a los datos.

Según lo anterior, el presente documento busca analizar el cómo son aplicados algoritmos de ML en la elaboración de modelos de score creditico. Esto ya que hay pocos documentos que aborden este tema y que expliquen de forma detallada la implementación. Para dicha labor se cuenta con una base de datos proveniente del periodo comprendido entre 1973 y 1975, la cual corresponden a un banco regional del sur de Alemania que cuenta con 700 créditos cuya obligación fue saldada y 300 créditos cuya obligación se incumplió.

En este trabajo se presentan dos modelos de score crediticio utilizando los algoritmos de ML: LR y SVM. Se explican los datos con los que se cuenta y se aplican los modelos de ML, para finalmente realizar una evaluación de los resultados obtenidos con cada uno utilizando la curva ROC y el AUC, también se mide la significancia estadística de las variables en LR.

Para ello se cuenta con una parte metodológica, la cual está conformada por la explicación de los datos, el desarrollo teórico de ambos modelos y el desarrollo de las medidas para cuantificar los mismos. Así mismo se tiene un apartado con los resultados obtenidos y por último se presentan las conclusiones. El trabajo busca que el lector comprenda el uso de algoritmos de ML, su aplicación para la predicción y clasificación en un caso real como lo es score crediticio.

Metodología

La base de datos para el trabajo fue obtenida del repositorio de bases de datos para Machine Learning de la Universidad de California en Irvine. El nombre de la base de datos es South German Credit, y ha sido editada por Grömping (2019), ya que la base original es *German Credit Data* la cual posee algunos errores debido a una mala traducción del alemán Grömping (2019). Un factor para que haya pocas investigaciones respecto a la implementación de credit scoring, es la falta de datos al ser información confidencial.

Según Grömping (2019) los datos presentes en la tabla fueron recolectados en el periodo de 1973 a 1975 y presentados por Häußler, Fahrmeir y Hamerle. Estos corresponden a un banco regional del sur de Alemania (cuyo nombre no se da a conocer), en total cuenta con 1000 créditos, en la base se muestra

cuando un contrato crediticio fue cumplido (bueno, good) o si fue incumplido (malo, bad). La cantidad de créditos buenos es 700 y malos 300. Los autores originales de la tabla afirman que los créditos malos están sobre representados, esto para tener suficiente información para discriminar entre buenos y malos.

La cantidad de variables de esta base de datos es de 21:

- laufkont(Status): Estado de la cuenta corriente del deudor.
- laufzeit(Duracion): Duración en meses del crédito.
- moral(Historial_crediticio): Cumplimiento con créditos anteriores o actuales.
- verw(Proposito): Motivo por el que deseaba el crédito.
- hoehe(Monto): Monto del crédito en Deutschmark.
- sparkont(Ahorros): Ahorros.
- beseit(Tiempo_trabajando): Tiempo que lleva en el trabajo actual.
- rate(Porcentaje_salario): Porcentaje de los ingresos que representan los pagos del crédito.
- famges(Estado_civil_sexo): Información combinada del sexo y el estado civil.
- buerge(Otros_deudores): Existen deudores o fiadores para el crédito.
- wohnzeit(Residencia): El tiempo que ha vivido el deudor en la residencia.
- verm(Propiedad_valiosa): Propiedad con mayor valor del deudor.
- alter(Edad): Edad del individuo.
- weitkred(Otras_deudas): Posee algún otro tipo de deuda.
- wohn(Residencia): Tipo de residencia.
- bishkred(Creditos_banco): Número de créditos que tiene o ha tenido en el banco, incluido el que se clasificó.
- beruf(Empleo): Tipo de trabajo.
- pers(Personas_dependientes): Número de personas que dependen del deudor.
- telef(Telefono): Si existe un número de teléfono registrado.
- gastarb(Extranjero): Si el deudor es extranjero.
- kredit(Default): Indica si el deudor cumplió o no con lo establecido en el contrato de crédito. Con valores 0 malo y 1 bueno.

Cuando se expone un algoritmo de ML es necesaria una medida que permita definir su eficacia, para esto se cuenta con la curva ROC. Se explican la curva ROC y AUC basado en lo expuesto por Del Valle (2017). Para entender la construcción de la curva se deben tener claro los siguientes conceptos:

Verdaderos positivos (TP): Son los resultados positivos que han sido clasificados por el modelo como positivos. Falso positivos (FP): Son los resultados negativos que han sido clasificados por el modelo como positivos. Verdadero negativo (TN): Son los resultados negativos que han sido clasificados por el modelo como negativos. Falso negativo (FN): Son los resultados negativos que han sido clasificados por el modelo como positivos. Sensibilidad: Es la proporción con base en la capacidad del modelo de dar resultados positivos. Esta dada por $\frac{TP}{TP+FN}$. Especificidad: Es la proporción con base en la capacidad del modelo de dar resultados negativos $\frac{TN}{TN+FP}$.

Una vez establecidos estos conceptos, se tiende a establecer una medida que relacione la sensibilidad con la especificidad, para esto existen las curvas ROC la cual se compone de los pares ordenados (1-Especificidad, Sensibilidad), que son construidos al variar el modelo en algún parámetro. En

este trabajo para ambos modelos se cambia la probabilidad necesaria para ser considerado como bueno o malo (punto de corte).

Al construirse la curva ROC se puede graficar también la diagonal $y = x$ que se interpreta como lo que sucedería si no se aplica el modelo y en su lugar se lanza una moneda para tomar la decisión de cómo clasificar a un individuo. Por lo que al obtener un punto encima de esta se interpreta que es mejor emplear el modelo con esa probabilidad para tomar la decisión que solo lanzar una moneda, y obtener un punto debajo significa que en ese caso sería mejor no ejecutar el modelo.

Si bien la curva ROC permite tener una idea del comportamiento del modelo, es difícil establecer comparaciones con otros modelos. Para solucionar este problema se cuenta con el AUC, el cual, al ser un número, facilita la medición de la eficacia del modelo, entre más cerca de 1 es mejor.

En cuanto a los modelos implementados, SVM se basa en la idea de encontrar un semiespacio (w, b) tal que el vector explicativo es x_i y la variable de respuesta es y_i con $i = 1, .., m$ entonces se obtiene como resultado:

$$y_i = \text{sign}(\langle w, x_i \rangle + b), \qquad \forall i$$

Sin embargo, no siempre es posible encontrar un resultado tan ideal por lo que se permite cierto grado de error en lo que se llama SVM suave (Shalev-Schwartz y Ben-David, 2014). El modelo de máquinas de soporte vectorial puede ser complejo y extenso por lo que para más información se puede referir a estos autores. En el caso del modelo finalmente implementado es el siguiente:

Ecuación 1

$$\min_{\omega, b, \xi} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i \right)$$

sujeto a $y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i$, con $\xi \geq 0$. Donde ξ_i son los errores y C sirve como penalización para los errores. Este problema de optimización puede ser resuelto mediante métodos numéricos, más específicamente el algoritmo de iteración descenso del gradiente estocástico. Se puede penalizar de acuerdo con la representación de clase mediante:

$$C_k = C \cdot w_j \qquad w_j = \frac{n}{k n_j}$$

donde j es la clase, n es el número total de observaciones y k es el número de clases de tal forma que n_j es el número de observaciones de la clase j.

No obstante, no siempre es posible encontrar una solución para la *Ecuación 1*, al menos en la dimensión donde se encuentran las variables de respuesta, por ello se aumenta la dimensión permitiendo encontrar una separación más clara de las clases, en lo que se conoce como truco del kernel.

Definiendo kernel como $K(x, x') = \langle \psi(x), \psi(x') \rangle$ donde ψ es un mapeo a un espacio de Hilbert, se tiene el siguiente teorema.

Teorema: Si se asume ψ un mapeo de $\mathcal{X}$ el espacio donde se ubica x_i es decir las observaciones. Entonces existe un vector $\alpha \in R^m$ tal que $w = \sum_{i=1}^{m} \alpha_i \psi(x_i)$ y es solución para la *Ecuación 1* de tal forma que se obtiene:

Ecuación 2

$$\min_{\alpha \in R^m} \left(\frac{1}{2} \alpha^T G \alpha + C \sum_{i=1}^{m} \max\{0, 1 - y_i (G\alpha)_i\} \right)$$

con G la matriz tal que $G_{ij} = K(x_i, x_j)$, y para predecir la clase a la que pertenece una observación se emplea:

$$\langle w, \psi(x) \rangle = \sum_{i=1}^{m} \alpha_j \langle \psi(x_j), \psi(x) \rangle$$

Entre los kernels más conocidos y utilizados en este trabajo están los siguientes:

- Polinomial: $K(x, x') = \gamma(1 + \langle x, x' \rangle)^k$ con k el grado del polinomio.
- Gaussiano: $K(x, x') = e^{-\gamma \|x - x'\|^2}$.
- Lineal: $K(x, x') = \langle x, x' \rangle$.
- Sigmoide: $K(x, x') = \tanh \gamma \langle x, x' \rangle + r$ con $r \in R$.

Para todos los casos $\gamma > 0$, e indica el grado de dependencia de la muestra de entrenamiento.

Un aspecto importante es que este modelo de SVM no regresa probabilidades lo cual puede ser importante para distintos aspectos como puede ser la curva ROC y conocer cuál es la probabilidad de que la persona incurra en default o no. Para ello se puede usar lo propuesto por Platt y col. (1999) que define:

$$f(x) = h(x) + b$$

donde $h(x)$ es la *Ecuación 2* es el resultado de la máquina de soporte vectorial que es la distancia a la que se encuentra x_i del hiperplano y se convierten estas distancias en probabilidades mediante:

Ecuación 3

$$P(y = 1 | f) = \frac{1}{1 + \exp(Af + B)}$$

Para esto se estiman los parámetros A y B mediante máxima verosimilitud y validación cruzada de 5 iteraciones. Se realiza también un cambio en la base de entrenamiento cambiando y_i por $t_i = \frac{N_+ + 1}{N_+ + 2}$ si $y_i = 1$ y $t_i = \frac{1}{N_\mp 2}$ si $y_i = -1$, para $i = 1, .., m$ con N_+ la cantidad de 1's y N_- la cantidad de -1's. Esto se hace para evitar sobre ajustes.

Finalmente, para medir el grado de ajuste se estima la curva ROC para la predicción y se calcula el AUC como medida de ajuste. Una forma rápida de implementar este modelo es mediante el paquete de Python (3.10) Scikit Learn y la función SVC, para saber más se puede consultar Pedregosa y col. (2011).

En la implementación se ha separado la base de datos en dos bases una de entrenamiento con 80% de los datos y 20% la de prueba, ambas con la misma proporción de buenos y malos, se han escalado ambas bases para ello se ha estandarizado dado que según Ahsan y col. (2011) esto brinda mejores resultados con SVM.

Para poder determinar el C, el kernel y si se debe penalizar el tamaño de la clase se emplea validación cruzada, usando como medida el AUC y se toma el que mayor valor obtiene. Posterior a esto se toma C, el kernel y la penalización o no de la clase, esto de acuerdo con lo obtenido anteriormente, y se vuelve a realizar validación cruzada para determinar el parámetro γ que muestra mejores resultados. Por defecto la función SVC toma $\gamma = \frac{1}{n \cdot V}$ donde n es el número de variables y V es la varianza de la matriz de datos de entrenamiento que corresponden a las variables explicativas, por lo que se tomó en cuenta también este caso para comparar con otros posibles valores de γ en el conjunto $\left\{\frac{1}{100}, \frac{1}{90}, .., \frac{1}{10}, 1, 10, 20, \dots, 100\right\}$.

Finalmente se mide el grado de predicción con la curva ROC y el AUC, empleando la base de entrenamiento. Además, para realizar una predicción y así obtener una matriz de confusión no se emplean las probabilidades, simplemente se muestra el resultado que brindaría la *Ecuación 2.*

Hältuf (2014) realiza una labor similar en su tesis como lo es la estandarización, para las probabilidades emplea el margen funcional y emplea el kernel gaussiano tras determinar que este es el que se ajusta de mejor manera a su base de datos. Además, emplea el AUC de la curva ROC como medida de ajuste. En el caso de Chow (2007) no indica escalación de las variables, pero sí emplea como medida de ajuste el AUC.

En cuanto a LR se explica brevemente el algoritmo basado en lo expuesto por Salcedo (2021), si se desea profundizar un poco más se puede referir a este. Si se supone X es el vector de las variables predictoras y Y la variable de respuesta entonces se plantea:

$$p_\beta(X) = P(Y = 1|X) = \frac{1}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

exponiendo así que Y_i es una variable aleatoria tipo Bernoulli con media p_i. Los parámetros β_j son estimados mediante máximo verosimilitud estableciendo:

$$\mathcal{L}(\beta) = \prod_{i=1}^{m} p(X_i)^{Y_i}(1 - p(X_i)^{1-Y_i}) \Rightarrow \log\big(\mathcal{L}(\beta)\big) = l(\beta) = \sum_{i=1}^{m} -\log\big(1 + e^{X_i\beta}\big) + \sum_{i=1}^{m} Y_i(X_i\beta)$$

luego derivando respecto a β e igualándolo a 0 se obtiene:

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m} \frac{1}{1 + e^{X_i\beta}} e^{X_i\beta} X_i + \sum_{i=1}^{m} Y_i X_i = X^T\big(Y - p(X)\big) = 0$$

esto debe ser resuelto mediante métodos numéricos, en este caso particular Newton-Raphson. Posterior a esto se tiene el modelo ajustado por lo que bastaría hacer la predicción basado en la *Ecuación 3* y la probabilidad obtenida.

Para la selección del punto de corte se trabaja con el índice de Youden, pues según lo expuesto por Del Valle (2017), este método permite el ajuste del modelo con base a la mayor sensibilidad y especificidad que ofrece la LR. Este estimador se define por:

$$\text{índice de Youden} = S + E - 1$$

En el cual se aprecia que entre más cerca de 1 el índice, mejor será el ajuste del modelo. Aplicando para ello la función minimize_scalar del paquete scipy.optimize de Python, sobre el inverso aditivo del índice de Youden, para que se maximice el mismo.

En cuanto la implementación de la LR en este trabajo se utiliza la función LogisticRegression del paquete Scikit Learn de Python. Se emplea la misma base de entrenamiento y de prueba construida para SVM y se realiza la predicción tomando el mejor punto de corte basado en el índice de Youden como la probabilidad mínima necesaria para ser categorizado como bueno, esto para construir una matriz de confusión. A su vez para medir el grado de ajuste del modelo se emplea la curva ROC y el AUC.

Además, dado los p-valores para cada una de las variables, obtenidos de la regresión logística, se ha aplicado una prueba de hipótesis para verificar la significancia estadística de cada una de ellas, para esto se ha planteado una prueba de hipótesis en la que la hipótesis nula H_0 consiste en que la variable en cuestión no es de significancia estadística. Así, bajo un umbral del 0.05, si el p-valor es de la forma $p < 0.05$ se rechaza la hipótesis nula, implicando que la variable en cuestión es estadísticamente significante.

Si el lector así lo desea, el código de este trabajo puede ser consultado en Cervantes y col. (2022).

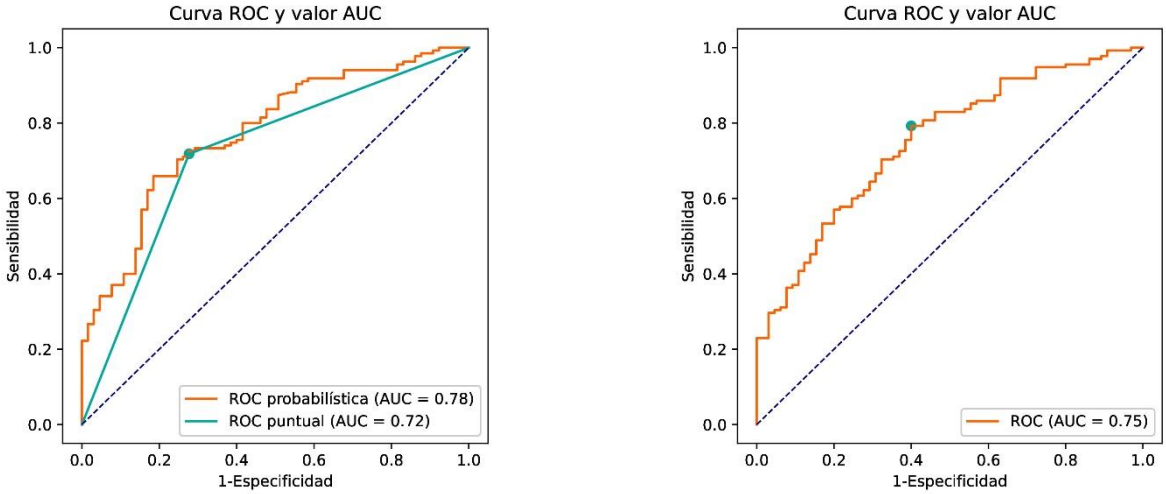
Resultados

Para SVM tras realizar la validación cruzada se determina que el mejor resultado se obtiene con $C = 1$, el kernel gaussiano, la penalización del tamaño de la clase y $\gamma = \frac{1}{70}$ que es distinto al predeterminado por SVC. Esto coincide con lo expuesto por Addo y col. (2018) al tenerse que el grado de representación de cada clase puede afectar en los resultados obtenidos por el modelo. A esto se le puede añadir que el manejo de los datos es de gran importancia como lo expone Innocenti (2019). Para LR se tiene que el punto de corte óptimo es 0.632.

En el caso de esta base de datos de no generarse de manera adecuada la muestra y no estandarizarse se obtiene resultados los cuales no son los más adecuados prediciendo que gran cantidad de los individuos malos son clasificados como buenos. Además, es importante ajustar los distintos parámetros del modelo tal y como lo recomiendan Pedregosa y col. (2011) y Meyer y col. (2022) y otros autores.

Figura 1

Respuesta de modelo LR y SVM



a) Fuente: Elaboración propia, datos de Grömping (2019).
b) Regresión logística (izquierda) y SVM (derecha).
c) Kernel gaussiano, $C = 1$ y $\gamma = \frac{1}{70}$.
d) LR: punto de corte 0.632.

Tabla 1

Matriz de confusión LR

		Predecido	
		V	F
Reales	V	107	28
	F	26	39

a) Fuente: Elaboración propia, datos del Grömping (2019).
b) Matriz construida con probabilidad 0.632 como decisión.

Tabla 2

Matriz de confusión SVM.

		Predecido	
		V	F
Reales	V	97	38
	F	18	47

a) Fuente: Elaboración propia, datos del Grömping (2019).
b) Kernel gaussiano y $C = 1$.
c) Matriz construida con SVM sin probabilidades.

En el caso de SVM se puede observar que el valor de AUC de la curva ROC es similar al obtenido por Hältuf (2014) y Chow (2007). A su vez se coincide en que es necesario escalar las variables y el kernel de mejor ajuste es el gaussiano para esta base de datos. Asimismo, se concuerda que, en cuanto al costo computacional, para obtener el kernel óptimo, puede ser alto. No obstante, la matriz de confusión muestra que se pudo haber predicho una gran cantidad de casos de default, sin embargo, el objetivo no es eliminar todos los casos de posible default ya que como expone Thomas (2009) el objetivo de las entidades financieras no es tener riesgo 0, ya que aquí no se encuentran los óptimos.

Se llega a una conclusión similar a la obtenida por Hältuf (2014) y Chow (2007) en cuanto al modelo que obtiene mayor grado de ajuste del AUC, SVM. A su vez se tiene un alto costo computacional. No obstante, SVM es capaz de predecir mayor cantidad de casos malos, lo cual puede ser relevante para una entidad financiera. Ahora tal y como exponen autores como Osisanwo y col. (2017) el modelo SVM es más novedoso, pero a su vez este es más complejo, por lo que se puede requerir de un personal más cualificado. Por lo que dependiendo de las necesidades y los recursos puede ser mejor emplear una LR antes que SVM, no obstante, en caso de ser posible se recomienda aplicar más de un modelo.

Por otro lado, los resultados obtenidos mediante la LR son consistentes con los obtenidos por estudios previos, en este sentido Al-Aradi (2014) utilizando la misma base de datos y empleando de igual manera un modelo de LR encuentra mediante la realización de un test de no aditividad, cuyos resultados sugieren una débil evidencia en contra del modelo, además en dicho estudio emplean un test de Hosmer-Lemeshow el cual encuentra poca evidencia en contra del ajuste del modelo. En este sentido si bien es cierto a las pruebas realizadas para cuantificar y medir la efectividad del modelo son distintas, se encuentra que empleando como método de ajuste la curva ROC se obtiene un ajuste similar a SVM y en general un poder predictivo de importancia relativa.

Por otro lado, aplicando la prueba de hipótesis comentada a la regresión logística, bajo los p-valores obtenidos para cada variable, se obtiene que algunas de ellas no tienen significancia estadística,

las cuales son: la residencia, la condición de extranjero, si cuenta o no con teléfono, el empleo, las personas dependientes, la edad y la variable otras deudas.

Conclusiones

El objetivo principal de esta investigación era cómo poder emplear algoritmos de ML para construir modelos de score crediticio. Esto se vio como una oportunidad para poder ver en acción el funcionamiento de algoritmos de ML, ya que es un tema el cual se encuentra ampliamente utilizado en la actualidad.

En esta investigación se pudo notar el potencial que tiene el algoritmo SVM para poder predecir, problemas que requieren clasificación binaria. No obstante, este puede llegar a ser complejo a nivel teórico. Teniéndose que de aplicarse de forma incorrecta pueden ser negativos más que beneficiosos en el manejo de riesgos.

De acuerdo con la metodología se puede construir un modelo de score crediticio con un algoritmo de ML de la siguiente forma:

- Seleccionar un modelo de ML de clasificación.
- Ajustar los datos para emplear este modelo, separar los datos en entrenamiento y prueba.
- Ajustar el modelo a los datos.
- Realizar la predicción y medir el grado de ajuste de la predicción en este caso el AUC.
- En caso de no obtenerse los resultados esperados se puede repetir a partir del paso 2, se puede considerar cambiar de modelo ML o una combinación de ambos.

El grado de ajuste de SVM, al menos en los resultados obtenidos no es tan superior con respecto a LR al menos bajo la AUC. No obstante, el primero es capaz de predecir una mayor cantidad de casos de default esto se puede notar en la matriz de confusión.

Durante el proceso se encontraron problemas por la forma en que habían sido tratados los datos para SVM, por lo que fue necesario escalar, también se vio la importancia de ajustar el parámetro C, γ y seleccionar un kernel adecuado. En futuros trabajos se puede considerar utilizar otros estimadores, se puede comparar el grado de ajuste con otras bases de datos y comparar los resultados con otros modelos.

Bibliografía

Addo, P. Guegan, D & Hassani, B. (2018). Credit Risk Analysis Using Machine and Deep Learning Models, *Risk, 6*(2).

Ahsan, M. Mahumud, M. Saha, P. Gupta, K. & Siddique, Z. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, *9*(3), 52. https://www.mdpi.com/2227-7080/9/3/52.

Al-Aradi, A. (2014). Credit Scoring via Logistic Regression. https://utstat.toronto.edu/~ali/papers/creditworthinessProject.pdf.

Banco Central de Costa Rica. (1998). *Acta de la sesión 4974-98.* https://bit.ly/38DgNX5.

Cervantes, J. Monge, M. & Sabater, D. (2022). *SVM y LR en score crediticio.* https://github.com/Afr063426/Proyecto_Est_Act_II.

Chow, J. (2007). *Analysis of Financial Credit Risk Using Machine Learning* (Tesis de maestría). Universidad de Aston.

CONASSIF. (2017). Propuesto de reglamento sobre gestión de riesgos de crédito.

Del Valle, A. (2017). *Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones* (Tesis de maestría). Universidad de Sevilla.

Grömping, J. (2019). *South German Credit* [Data set]. https://bit.ly/3Lsf0RV.

Grömping, J. (2019). South German Credit Data: Correction a Widely Used Data Set. *Reports in Mathematics, Physic and Chemistry, 4.* http://www1.beuth-hochschule.de/FB_II/reports/Report-2019-004.pdf.

Gutiérrez, M. (2007). Credit scoring models: what, how, when and for what purposes. *Banco Central de la República Argentina. Munich Personal RePEc Archive*. Octuber, 1-32.

Hältuf, M. (2014). *Support vector machines for credit scoring* (Tesis de maestría). Universidad de Económicas en Praga.

Innocenti, F. (2019). Machine Learning in Credit Scoring.

Ławrynowicz, A. & Tresp, V. (2014). Introducing machine learning. *Perpectives on Ontology Learning; Lehmann, J. , Voelker, J., Eds,* 35-50.

Meyer, D. Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F. Chang, C. & Lin, C. (2022). *Package e1071.* https://cran.r-project.org/web/packages/e1071/e1071.pdf.

Osisanwo, F. Akinsola, J. Awodele, O. Hinmikaiye, J. Olankanmi, O. & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trendsd and Technology*, *48*, 128-138. https://bit.ly/3vwGXSO.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learni: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, *10*(3), 61-74. https://bit.ly/3PpdfYd.

Salcedo, P. (2021). *Modelo de regresión logística.*

Shalev-Schwartz, S. & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

Thomas, L. (2009). *Consumer Credit Models: Pricing, Profit and Portfolios.* Oxford University Press.

Wehle, H. (2017). *Machine Learning, Deep Learning and AI: What's the Difference? https://www.researchgate.net/publication/318900216_Machine_Learning_Deep_Learning_and_AI_What%27s_the_Difference*