UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Facultad de Administración y Dirección de Empresas /UPV

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Facultad de Administración y Dirección de Empresas

Diseño e implementación de un modelo de cuantificación de riesgo de crédito de una entidad bancaria.

Trabajo Fin de Grado

Grado en Administración y Dirección de Empresas

AUTOR/A: García-España Simó, Pablo

Tutor/a: Moya Clemente, Ismael

CURSO ACADÉMICO: 2022/2023

AGRADECIMIENTOS

Quiero agradecer la dedicación de mi tutor Ismael Moya en este TFG, en el que me ha orientado y apoyado a lo largo de todo el proceso. Agradezco de corazón a mi familia que siempre están ahí, ayudándome en cada paso que doy.

Agradecer también la orientación de mi amigo Miguel y Jaime que tanto conocimiento me han aportado en este TFG. A mis amigos de la universidad, en especial Rafa y Joselu, que tanto me han hecho disfrutar de esta etapa universitaria.

RESUMEN

El presente trabajo tiene como objeto el diseño e implementación de un modelo de cuantificación de riesgo de crédito de una entidad bancaria. Para ello se hace uso de una extensa base de datos de Kaggle y la herramienta de programación Python, que, mediante el uso de la serie de datos y variables, se elabora un código que genera un modelo entrenado capaz de decidir la concesión o no del crédito mediante una técnica denominada regresión logística. También se realiza un "Dashboard" el cual permite introducir manualmente las variables que se deseen de forma que se puede interactuar con el modelo para que éste indique el resultado.

El enfoque principal es analizar las distintas variables que intervienen en estos créditos, y el peso que tienen, mejorando así la comprensión de cómo los bancos conceden los créditos. Se realiza para ello una revisión del estado del arte, así como de numerosos conceptos clave que son de ayuda para complementar el TFG. Además, analizar el riesgo de crédito, es un elemento clave para tomar decisiones para cualquier entidad financiera, una gestión inadecuada del mismo puede derivar en problemas de estabilidad económica.

Palabras Clave: Python, riesgo de credito, entidad bancaria, modelo IA.

RESUM

El present treball té com a objecte el disseny i implementació d'un model de quantificació de risc de crèdit d'una entitat bancària. Per a això s'utilitza una extensa base de dades de Kaggle i l'eina de programació Python, que, mitjançant l'ús de la sèrie de dades i variables, es desenvolupa un codi que genera un model entrenat capaç de decidir la concessió o no del crèdit mitjançant una tècnica anomenada regressió logística. També es crea un "Dashboard" que permet introduir manualment les variables desitjades de manera que es pugui interactuar amb el model perquè aquest indiqui el resultat.

L'enfocament principal és analitzar les diferents variables que intervenen en aquests crèdits i el pes que tenen, millorant així la comprensió de com els bancs concedeixen els crèdits. Es realitza, per a això, una revisió de l'estat de l'art, així com de nombrosos conceptes clau que són d'ajuda per complementar el projecte. A més, analitzar el risc de crèdit és un element clau per prendre decisions per a qualsevol entitat financera, una gestió inadequada del mateix pot derivar en problemes d'estabilitat econòmica.

Paraules clau: Python, risc de crèdit, banc, model d'IA.

ABSTRACT

The present work aims to design and implement a credit risk quantification model for a banking institution. To achieve this, an extensive Kaggle database and the Python programming tool are used. By using the data series and variables, a code is developed, and it generates a trained model capable of deciding whether to grant credit or not through a technique called logistic regression. A "Dashboard" is also created, which allows manually entering the desired variables, enabling interaction with the model to provide the outcome.

The focus is to analyse the different variables involved in these credits and their respective weights, thereby improving the understanding of how banks grant credits. A review of the state of the art is conducted, as well as numerous key concepts that help complement the project. Furthermore, analysing credit risk is a crucial element in making decisions for any financial institution, as inadequate management can lead to economic stability issues.

Keywords: Python, credit risk, bank, AI model.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

OBJETIVO Y JUSTIFICIACIÓN DEL TFG

El presente trabajo tiene como objeto el diseño, elaboración y análisis de un modelo de riesgo de crédito bancario. Gracias a la elaboración de dicho modelo mediante el uso de la base de datos de Kaggle, herramientas de Machine Learning y Python, se conocerá en profundidad como manejar el riesgo de crédito, así como las distintas variables que determinan la viabilidad de los créditos y como estas determinan su éxito o fracaso.

Al tener la posibilidad de elaborar un programa el cual permitirá poder ver y tocar las variables que determinarán el resultado gracias al modelo entrenado, será un TFG práctico y motivador en el que quedará confeccionada una herramienta muy útil e innovadora. Tradicionalmente, la solvencia del deudor para determinar el riesgo de recuperar la inversión del prestamista iba directamente ligada a los ingresos del deudor, sin embargo, gracias al uso de la tecnología ahora se tienen en cuenta más variables, aparentemente se ha simplificado mucho su cálculo, y además gracias al *Machine Learning* para el manejo del *Big data,* permite conocer para la entidad de financiación si conforme al histórico, es conveniente concederle o no el crédito.

Además, analizar el riesgo de crédito, es un elemento clave para tomar decisiones para cualquier entidad financiera, una gestión inadecuada del mismo puede derivar en problemas de estabilidad económica. Toda entidad de financiación dispone de un modo u otro de un modelo de riesgo de crédito, se trata de un entorno donde existe una gran competitividad por lo que cuanto más preciso y eficiente sea su modelo, mayor ventaja económica tendrá.

MOTIVACIÓN DEL TFG

La motivación de mi TFG surge por mi interés en el sector financiero, elaborar un TFG realista y práctico para la gestión de riesgos de crédito es un TFG vanguardista y con una tremenda aplicación práctica el cual me permitirá adquirir conocimientos acerca de la concesión de créditos y cómo manejar el riesgo de cada uno de ellos.

Como futuro graduado en economía, supone también un desafío personal, realizar un TFG de envergadura real, el cual, podría servir para una entidad de financiación. Se partirá de un estudio y unos conocimientos teóricos, alcanzando resultados reales y tangibles. Además, dado que será realizado mediante Python, servirá para reforzar conocimientos de programación y de aprendizaje de máquinas, desarrollando así un modelo entrenado que será la herramienta principal del TFG.

Por otro lado, el uso del Big Data y las herramientas de Machine Learning se están convirtiendo en un factor clave para una gestión adecuada del riesgo, tener la posibilidad de profundizar en el uso de estas tecnologías y como se aplican al ámbito financiero me permitirá realizar un TFG que combina justamente la titulación cursada, ampliando así los conocimientos en la rama de Administración y Dirección de Empresas y de ingeniería de Telecomunicaciones.

METODOLOGÍA

Una vez definido el objetivo del TFG, se elaborará una hoja de ruta de las distintas tareas que servirán de guía y ayuda para realizar el trabajo:

- Estado del arte: En este apartado se realizará una introducción al marco teórico del trabajo, ubicándonos en el momento actual del riesgo de crédito tras realizar una breve revisión de la historia del crédito.
- Conceptos de riesgo de crédito, modelos, Machine Learning…: Aquí se definirán una serie de conceptos clave para la comprensión del TFG y aumento del conocimiento tanto a nivel crediticio como tecnológico.
- Obtención de la base de datos de Kaggle: Se explicará cómo se ha obtenido la base de datos, así como los elementos que la componen, su origen y su posterior manejo.
- Diseñar el modelo, definiendo qué técnica estadística se utilizará: Se elegirá la técnica estadística que se utiliza en la elaboración del modelo, así como su justificación.
- Transformación y limpieza de datos para homogeneizar la base de datos: Se realizará una adaptación de los datos para tenerlos homogeneizados y poder así trabajar con ellos.
- Programar el modelo: Programación del modelo que permita obtener el objetivo inicial, se explicará su código realizado en Python.
- Analizar los resultados y elaborar las conclusiones: Por último, se realizará un breve análisis de lo obtenido con respecto a la realidad, así como una serie de conclusiones en las que se propondrán recomendaciones de mejora del modelo.

CAPÍTULO 1: ESTUDIO TEÓRICO

1.1 ESTADO DEL ARTE

La palabra "crédito" viene del latín, y quiere decir "creer" o "confiar". En la antigüedad, el uso de monedas y registros escritos permitió el surgimiento del concepto de deuda, lo que permitía cuantificar los bienes y servicios que se debían a los poseedores de monedas. Además, el dinero pasó a ser una forma de llevar cuentas de lo que se debía a cada persona y se simplificaba así el intercambio de bienes y servicios en la sociedad. Originalmente, el crédito se basaba en la confianza mutua entre las partes involucradas, pero con el tiempo, los prestamistas comenzaron a cobrar intereses por prestar el dinero. Ya en la época de los romanos, el crédito con intereses no estaba regulado por ninguna autoridad y podía generar tasas exorbitantes para los prestatarios, incluso a veces perdiendo su propiedad o siendo esclavizados para pagar sus deudas. (Gutiérrez, I., 2022)

Con la llegada del cristianismo a Europa, se consideró pecado cobrar intereses por los préstamos, lo que generó un fuerte sentimiento antisemita porque ellos eran los únicos que siguieron realizando el cobro de intereses. Sin embargo, con el tiempo, algunas familias cristianas también comenzaron a hacer préstamos con intereses y se convirtieron en banqueros. Con el auge de las grandes empresas comerciales y las guerras en Europa en el siglo XVII, surgieron varias entidades bancarias como el Banco de Suecia y el Banco de Inglaterra que tenían un gran tamaño, los cuales se convirtieron en pioneros en la banca moderna y comenzaron a prestar dinero con contratos y una tipo de interés fijo, en un horizonte temporal determinado.

En el siglo XIX en los Estados Unidos, hubo un período conocido como "banca libre" en el que los bancos no estaban muy regulados y podían emitir dinero sin tener la cantidad necesaria de respaldo. El Banco de América del Norte, fue el primero comercial del país y financió la Guerra de la Independencia (Olegario, R., 2019). Durante el siglo XX hubo un gran aumento del número de bancos, los cuales comenzaron a financiar un gran número de operaciones debido a la mejora económica que experimentaba el país. Los fabricantes fueron la principal fuente de la financiación a plazos y los bancos vieron que el comportamiento de los consumidores cambió, estaban dispuestos a pagar cantidades de dinero muy superiores en caso de que pudieran pagar a plazos. Tanto era así, que al final de la Primera Guerra Mundial, el 25% de las familias dependía de los créditos a plazos en comparación con apenas un 11% en 1880 y debido a la creciente disponibilidad de pequeños préstamos, la deuda de los hogares en Estados Unidos alcanzó los 880 dólares cuando el salario anual era de 475 dólares. (Calder, 1999).

El negocio de los préstamos se multiplicó por 25 en los primeros 40 años de los '90, y no disminuyó hasta después de la Segunda Guerra Mundial que vino acompañada de regulaciones y permitió un mayor ahorro en las viviendas.

En general, hoy en día, los bancos del mundo tienen la obligación de tener una reserva fraccionaria de mínimo el 5% o 10% de lo depositado por los ahorradores. Esta medida se utiliza para prevenir que se especule a nivel financiero y exista una creación sin límites de dinero sin nada detrás. Gracias a la reserva fraccionaria, los bancos pueden crear dinero y estimular el desarrollo económico de una sociedad. No obstante, emitir una excesiva cantidad de crédito puede llevar a la aparición de grandes crisis financieras. (Olegario, R., 2019)

Un ejemplo de esto es la crisis financiera de 2008 en Estados Unidos, que generó la Gran Recesión. La Reserva Federal redujo las tasas de interés para estimular el desarrollo económico, pero algunos bancos tradicionales usaron ese dinero para emitir dinero a bajo coste a personas que querían comprar una casa, incluso aunque fueran personas con deudas y de alto riesgo. Los bancos dieron grandes préstamos de dinero a personas que no podían hacer frente al pago y desarrollaron instrumentos financieros más complejos como las hipotecas subprime. Cuando los inversores vieron lo que las hipotecas subprime eran en realidad, y que no cobrarían su dinero, vendieron masivamente provocando el desastre.

La crisis financiera de 2008 llevó al gobierno de Estados Unidos a regular el sistema financiero y promulgar la ley Dodd Frank con el objetivo de dar protección a los consumidores financieros de forma que se comenzó a llevar un control más exhaustivo de cómo se concedían los créditos.

El microcrédito se ha convertido en uno de los elementos más utilizados del mundo bancario y crediticio moderno (Gutiérrez, I., 2022). El empresario Muhammad Yunus fundó la Fundación Gramen Bank y popularizó este producto financiero a fines del siglo XX. Tenía como objetivo hacer que el crédito sea más accesible para las personas con menos fondos y menos seguridad. Estos pequeños préstamos, tenían tasas de interés un poco más elevadas que las de los bancos tradicionales y han mejorado las oportunidades de vida de aquellos más pobres. Antes de la llegada del microcrédito, los más pobres no podían dar una garantía de pago a los grandes bancos, por lo que no podían obtener crédito. En 2006, Yunus Bank y Gramen Bank recibieron el Premio Nobel de la Paz por su colaboración en contra de la pobreza a través de microcréditos.

Hoy, el microcrédito es omnipresente en todas partes y varias instituciones financieras, incluidos los neobancos, están avanzando en proyectos que promueven una banca más inclusiva. En los últimos años, la aparición de novedosos instrumentos financieros como lo son las criptomonedas (2008) y los activos digitales como las NFT (2020) han provocado un replanteamiento del papel de los bancos en el mundo moderno. Hoy en día, los bancos tradicionales ya no tienen el monopolio de los préstamos, y muchos pueden pedir prestado a entidades que no son convencionales tal como los intercambios de criptomonedas y los

neobancos. Son precisamente estas últimas empresas las que plantean el mayor reto a la banca tradicional: asignar fondos de manera eficiente. Ante esta situación, muchos bancos han comenzado a cambiar sus servicios a un enfoque más digital y han establecido filiales de instituciones financieras que operan completamente en línea.

Estas empresas están motivadas por la innovación y buscan utilizar nuevas técnicas matemáticas para distinguir entre clientes. Entre estas técnicas se incluyen el uso de Big Data, datos no comunes y métodos de aprendizaje automático. En la actualidad, estas tecnologías están marcando un antes y un después en el entorno financiero, de hecho, los ratings de crédito entre las instituciones financieras tradicionales, como el FICO score, y los nuevos análisis han pasado de tener una coincidencia del 80% en el año 2007, a tan solo un 35% en el año 2015, tal y como señalan Jagtiani y Lemieux en su estudio (2019).

El presente trabajo tiene su desarrollo del modelo justo después de la crisis financiera de 2008, en el año 2011, fecha de la que data la base de datos que se utilizará para el desarrollo de la red neuronal para el aprendizaje. La base de datos proviene de Kaggle, una importante fuente de bases de datos que se utilizan para el desarrollo de modelos de redes neuronales, Kaggle es una empresa perteneciente a Google, siendo en este caso además una base de datos que dio origen a una competición de creación de redes neuronales para entrenar modelos, por lo que se ha utilizado una fuente fiable y constatada para la realización del modelo.

1.2 CONCEPTOS BÁSICOS DEL RIESGO DE CRÉDITO

En finanzas, es importante entender el concepto de riesgo de crédito, lo que requiere definir los términos "riesgo" y "crédito". El riesgo se refiere al conjunto de probabilidad de que ocurra un evento y sus efectos adversos.

La Real Academia Española (RAE) define el riesgo como contingencia o daño inminente. Un crédito, por su parte, es una cantidad o valor equivalente que una persona o entidad debe a otra y que puede ser reclamada y cobrada por un acreedor. El riesgo de crédito o riesgo crediticio, por tanto, se refiere a la pérdida potencial para un acreedor si un deudor incumple total o parcialmente una transacción financiera o comercial. Teniendo esto en cuenta, es importante considerar esto a la hora de conceder un préstamo, ya que puede afectar significativamente a la solvencia y estabilidad financiera del prestamista.

El riesgo de crédito se calcula como cualquier otro tipo de crédito y se basa en fórmulas estandarizadas (Calzada, A. 2020). En esto, el Comité de Supervisión Bancaria de Basilea es la entidad más reconocida en la definición de estándares y metodologías para la evaluación del

riesgo de crédito y define las fórmulas más utilizadas para el cálculo del riesgo de crédito, una de las fórmulas más utilizadas es la [1]

$$PE = PD \times EAD \times LGD\ (1-R) \qquad [1]$$

Esta fórmula, nos permite calcular la pérdida esperada (PE) asociada al préstamo. La fórmula contiene varias variables como: Probabilidad de Impago (PD), indica la probabilidad de que el deudor no cumpla con sus obligaciones contractuales. La exposición en caso de impago (EAD) se refiere al valor de una posición en caso de impago, mientras que la pérdida en caso de impago (LGD) mide la pérdida para los acreedores y se basa en las tasas de recuperación (1-R). Esta fórmula es una herramienta fundamental para evaluar el riesgo de crédito.

El riesgo de crédito puede clasificarse en dos tipos, minorista y mayorista, en función del prestatario. El primero está relacionado con la financiación de particulares y PYMES, como hipotecas, consumo y tarjetas, y el segundo con derivados de actividades comerciales relacionados con ventas y fusiones/adquisiciones. Calzada, A. (2020).

Asimismo, el riesgo de crédito se puede clasificar en función de su origen en cuatro tipos:

1. El primero es el riesgo de impago que surge cuando el deudor incumple total o parcialmente sus obligaciones.
2. El segundo es el riesgo de rebaja, sucede cuando se produce una rebaja de la deuda.
3. El tercer tipo es el riesgo de exposición. Se refiere a un importe pendiente incierto debido a las condiciones del mercado o a la propia empresa del deudor.
4. Por último, el riesgo de diferencial se centra en los rendimientos de los activos de riesgo con respecto a los que son de libre riesgo.

De acuerdo con la Circular 4/2016 del Banco de España, hay varios tipos de riesgo:

1. Riesgo normal.
2. Riesgo normal en vigilancia especial: operaciones con más riesgo de lo habitual.
3. Riesgo dudoso por razones distintas a la morosidad del titular, existen dudas de recuperar todo el dinero.
4. Riesgo fallido: Se considera poco posible recuperar el dinero.

Para gestionar el riesgo de crédito, (Peterdy, K., 2023) en términos generales se puede hacer en 2 pasos:

- **Medición**: El riesgo de crédito lo miden los prestamistas mediante herramientas de clasificación de riesgo patentadas, que en función de la empresa y el deudor varían. Los préstamos personales son más sencillos de determinar y serán el foco de este trabajo, estos, tienden a depender de una garantía personal y colateral, lo que significa que en caso de no cumplir se podría reclamar por parte de la entidad lo que se puso como aval del deudor para que sirva como liquidación de su deuda.

  En el caso de los préstamos comerciales son mucho más complejos ya que normalmente solicitan cantidades de dinero mayores y hay que evaluar el entorno empresarial, la industria, el negocio en sí y comprender la reputación de la empresa, así como al equipo directivo.

  Según las técnicas de análisis, los modelos y los parámetros de suscripción de propiedad exclusiva del prestamista, la evaluación crediticia de un prestatario determinará una puntuación. (Peterdy, K., 2023)

  La puntuación puede llamarse de distintas maneras. Por ejemplo, las puntuaciones de los instrumentos de deuda pública se denominan calificaciones crediticias o calificaciones de deuda (es decir, AAA, BB+, etc.); para prestatarios personales, pueden llamarse calificaciones de riesgo (o algo similar).

- **Mitigación**: El riesgo crediticio, si no se mitiga adecuadamente, puede resultar en pérdidas crediticias para un prestamista; las pérdidas afectan negativamente a la rentabilidad de las empresas de servicios financieros. Algunos ejemplos de estrategias que utilizan los prestamistas para mitigar el riesgo de crédito (y la pérdida de préstamos) incluyen, entre otros:
  - **Estructura crediticia:** Incluyen el período de amortización, el uso de (y la calidad de) la garantía colateral, LTV (préstamo a valor) y convenios de préstamo, entre otros.
  - **Análisis de sensibilidad:** El prestamista cambia ciertas variables en la estructura crediticia propuesta para ver cómo sería el riesgo crediticio del prestatario si las condiciones hipotéticas se hicieran realidad
  - **Controles a nivel de cartera:** Seguimiento y la comprensión de qué proporción de la cartera total de préstamos es un tipo particular de crédito o qué proporción de los prestatarios totales tienen una determinada puntuación de riesgo.

- **Modelos de riesgo:** Se explicarán posteriormente

Es importante destacar que para que se le conceda el crédito los bancos tienen en cuenta antes de ponerse a analizar si quiera (Fernández, L., 2023) el cumplimiento de cuatro requisitos fundamentales:

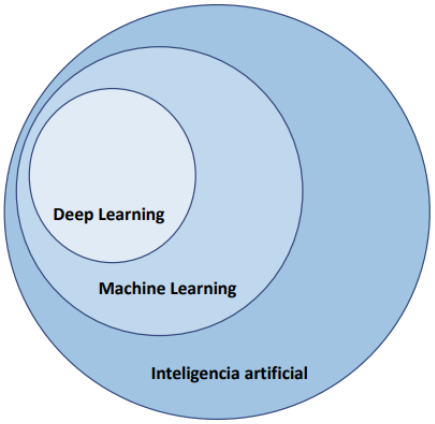
- Comprobación de los ingresos netos del deudor, de forma que la cuota de devolución del crédito no suponga más del 30% de sus ingresos netos.
- Disponer de estabilidad laboral, por lo que las personas que disponen de contratos temporales o autónomos no tienen la misma facilidad de obtenerlo que un indefinido.
- Tener un historial bancario positivo, se valora el haber cumplido con las obligaciones de pago pasadas.
- Lista RAI/ASNEF de morosidad, en caso de estar en estas listas por haber incurrido en impagos, no se te concederán los créditos y préstamos. Si que se podrían solicitar microcréditos por empresas externar a intereses superiores.

1.3 CONCEPTOS BÁSICOS DE INTELIGENCIA ARTIFICIAL Y MACHINE LEARNING

La Inteligencia Artificial es un conjunto de técnicas y algoritmos que son usadas para solventar cuestiones que el ser humano es capaz de solventar por intuición, pero que para la tecnología es complicado. Estos procesos incluyen el aprender, razonar y autocorregirse. (Albiol, A., 2022)

En las figura 1, se muestra las diferencias que existen entre IA, ML y DL, las cuales están integradas una dentro de la otra.

Figura 1. Dimensiones IA.



FUENTE: ALBIOL. A. (2022)

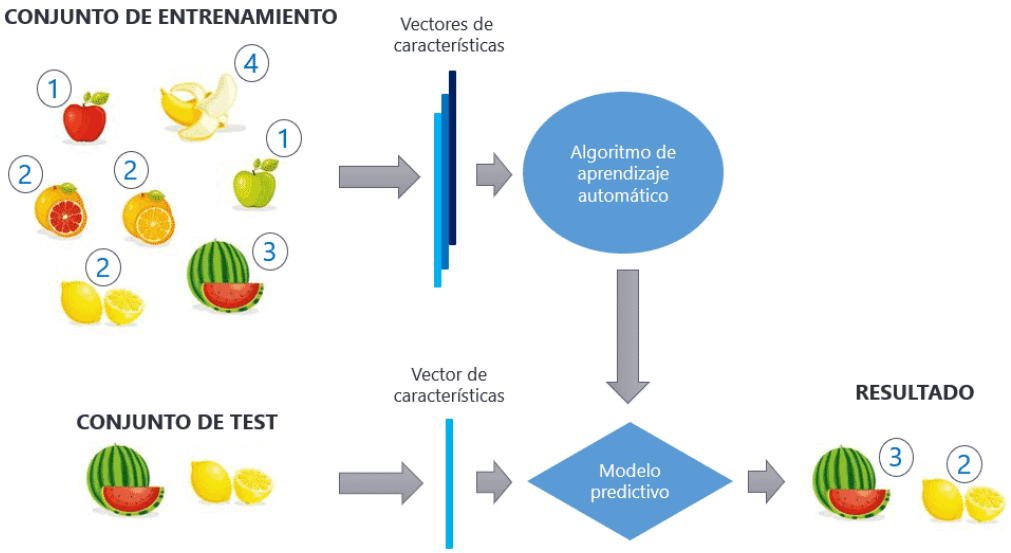
El Machine Learning es una disciplina que forma parte de la inteligencia artificial. Dentro del campo del aprendizaje automático, se encuentra el Deep Learning, el cual se basa en el uso de redes neuronales como elemento fundamental en sus algoritmos. Una característica distintiva del Deep Learning es la presencia de múltiples capas de nodos en las redes neuronales, lo que le otorga una mayor profundidad. Es importante mencionar que un algoritmo de Deep Learning debe contar con más de tres capas para ser considerado como tal. Kavlakoglu, E. (2020).

Los modelos de aprendizaje automático se dividen en tres tipo principalmente.

- El aprendizaje **supervisado**, también conocido como aprendizaje automático supervisado, se caracteriza por utilizar conjuntos de datos etiquetados para entrenar algoritmos que son capaces de clasificar datos o predecir resultados de manera precisa. Durante el entrenamiento, a medida que los datos de entrada son presentados al modelo, este ajusta los pesos hasta lograr una configuración adecuada.

  En la figura 2, se muestra a nivel visual de cuál es el método de funcionamiento de este aprendizaje:

Figura 2. Ejemplo aprendizaje supervisado

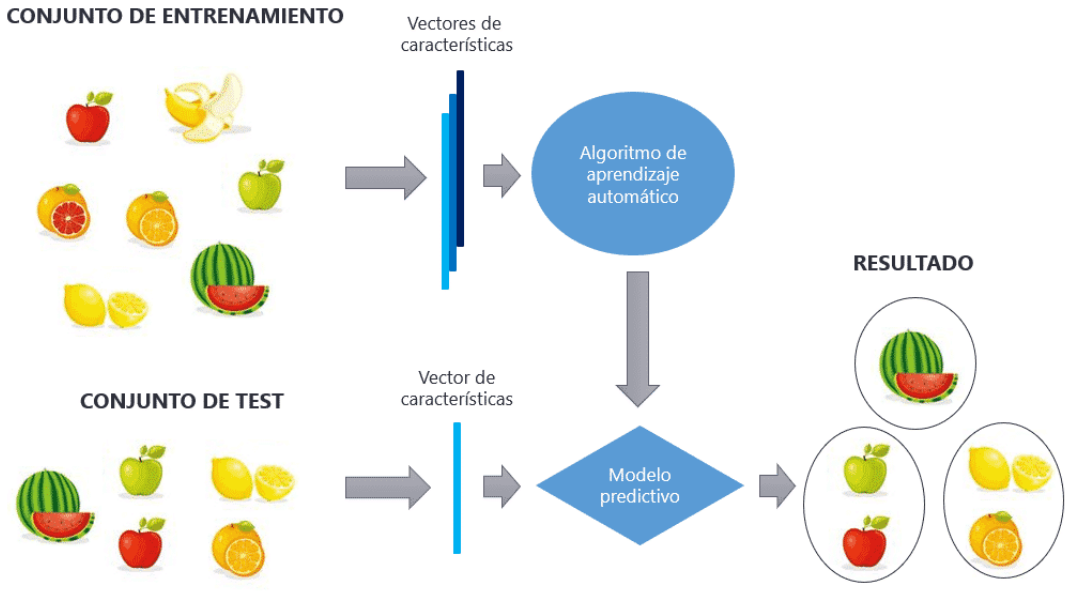


Fuente: Calvo, D. (2019)

- El aprendizaje **no supervisado**, que se conoce como aprendizaje automático, utiliza algoritmos para aprender de forma automática, analizando y agrupando conjuntos de datos no etiquetados. Los algoritmos identifican patrones ocultos o conjuntos de datos sin ayuda humana.

  Del mismo modo, y para el mismo caso de la fruta, se muestra en la figura 3, el funcionamiento del no supervisado:

Figura 3. Ejemplo aprendizaje no supervisado



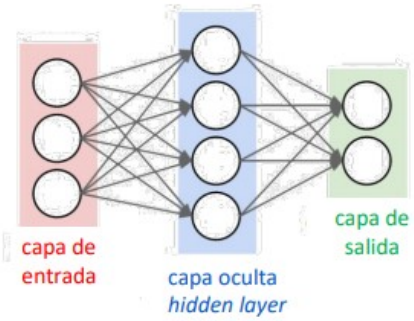
Fuente: Calvo, D. (2019)

- El aprendizaje **semi-supervisado** proporciona una solución intermedia entre el aprendizaje supervisado y el no supervisado. Para el entreno, emplea una serie de datos etiquetados más reducida que guía la clasificación y extrae las características de un conjunto de datos más grande que no está etiquetado. El aprendizaje semi-supervisado puede solventar el inconveniente de no tener bastantes datos etiquetados para un algoritmo de aprendizaje supervisado.

A diferencia de los algoritmos de aprendizaje no supervisado, los algoritmos de aprendizaje supervisado se basan en el uso de datos etiquetados. Estos algoritmos utilizan esos datos para predecir resultados futuros o asignarlos a categorías específicas, según el tipo de problema de regresión o clasificación que se esté abordando. Aunque los algoritmos de aprendizaje supervisado tienden a ser más precisos, requieren la ayuda inicial de los seres humanos para etiquetar los datos de manera adecuada. Sin embargo, esta etiquetación de datos facilita la computación, ya que no es necesario contar con un gran conjunto de datos de entrenamiento para obtener los resultados deseados. Algunas técnicas comunes de regresión y clasificación en el aprendizaje supervisado incluyen la regresión lineal y logística, el algoritmo naïve Bayes, el algoritmo KNN y el bosque aleatorio. **Estas técnicas serán explicadas en el siguiente punto, junto con los modelos de riesgo.**

Las redes neuronales, y más en concreto, las redes neuronales artificiales (ANN), copian la actuación del cerebro humano a través de una serie de algoritmos. Son un conjunto de capas compuestos por neuronas, tal y como se muestra en la figura 4. (Albiol, A., 2022)

Figura 4. Fases red neuronal.



Fuente: Albiol, A. (2022)

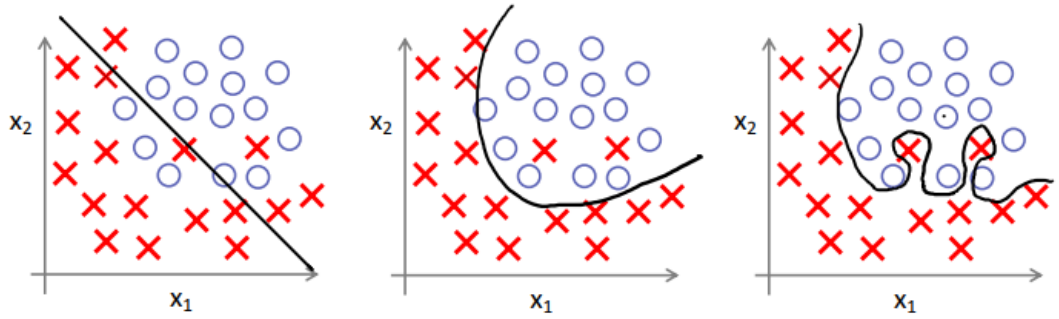
Las redes neuronales artificiales (ANN) están formadas por capas de nodos que incluyen una capa de entrada, una o varias capas ocultas y una capa de salida. Cada nodo, o neurona artificial,

establece conexiones con otros nodos y se le asigna un peso y un umbral. Si la salida de un nodo supera el umbral establecido, el nodo se activa y transmite datos a la siguiente capa de la red. Si no se cumple esta condición, el nodo no transmite datos a la siguiente capa.

En el contexto del Deep Learning, el término "profundo" se refiere únicamente a la cantidad de capas en una red neuronal. Una red neuronal que consta de más de tres capas, incluyendo la capa de entrada y la capa de salida, se considera un algoritmo de Deep Learning o una red neuronal profunda. Por otro lado, una red neuronal con solo tres capas se considera una red neuronal básica.

Es también importante realizar un buen ajuste del modelo, en los problemas de regresión lineal y logística, se realiza de forma habitual una técnica de clasificación basada en el uso de polinomios, de forma que se aumenta o disminuye el grado de los polinomios para ajustar la línea a un conjunto de datos, los cuales no siempre se encuentran entre la mitad y mitad como puede verse en la figura 5.

Figura 5. Clasificaciones variables.



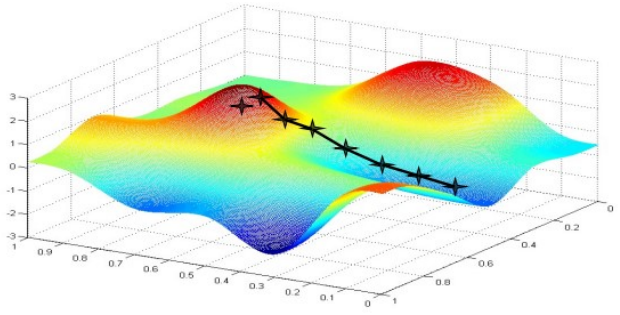
Fuente: Albiol, A. (2022)

Aumentar el grado del polinomio proporciona un mejor ajuste, sin embargo, si esto se sigue haciendo hasta llegar a la figura de la derecha, obtenemos lo que se llama un sobreajuste pues ante la entrada de nuevos datos no funcionará correctamente. De forma contraria ocurre en la de la izquierda, en el que no se ha proporcionado un ajuste adecuado por ser muy lineal.

Para minimizar el error, se realiza lo que se llama el descenso por gradiente, que se trata de un algoritmo que minimiza la función del error, para ello realiza una iteración ajustando en cada una de ellas la dirección del gradiente de descenso, el gradiente es la tasa de cambio de la función del error en una dirección concreta, de forma que los parámetros se ajustan en la

dirección opuesta al gradiente, disminuyendo así el error. En la figura 6, se muestra el funcionamiento gráfico del mismo.

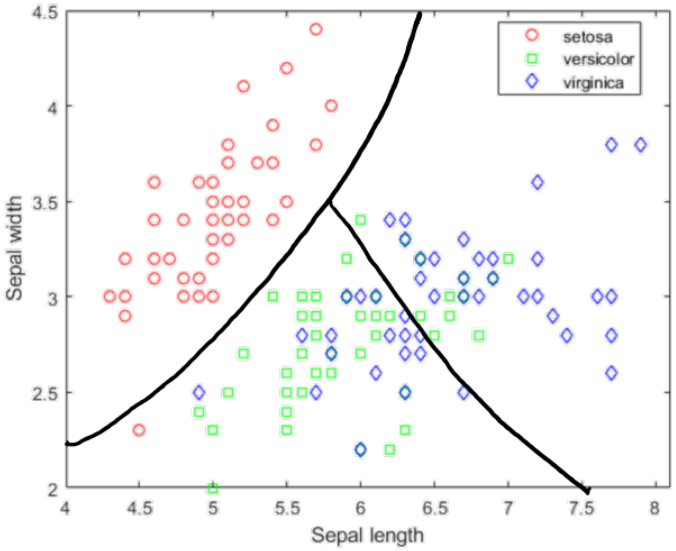
Figura 6. Descenso por gradiente.



Fuente: Albiol, A. (2022)

Por último, para evaluar el modelo se suele realizar una matriz de confusión. A continuación, se muestra un ejemplo de la asignatura Tratamiento Digital de la Señal, de la Universidad Politécnica de Valencia impartida por Alberto Albiol, en el que se pone como ejemplo un clasificador de setas en la figura 7.

Figura 7. Resultados ejemplo.



	Resultados del clasificador			
Groundtruth		Setosa	Versicolor	Virgínica
	Setosa	38	1	0
	Versicolor	0	31	12
	Virgínica	0	18	26

Fuente: Albiol, A. (2022)

De esta manera se puede realizar una evaluación del modelo tal y como realizaremos posteriormente en este trabajo. Para evaluar la precisión del clasificador de las setas se realiza la fórmula [2].

$$Accuracy = \frac{\sum Pred == GT}{Nsamples} \qquad [2]$$

De forma que en este caso sería la suma de 38, 31 y 26 dividido entre las muestras totales que proporcionaría una precisión del 75%.

1.4 MODELOS DE RIESGO DE CRÉDITO

El motivo principal por el que se crean los modelos para analizar el riesgo de crédito es para analizar cuánto capital se necesita para sostener las diferentes actividades de tomas de riesgo de una entidad bancaria. Este capital mínimo fue en primer lugar determinado conforme a Basilea I y que posteriormente evoluciono a Basilea II y detalló en Basilea III.

A medida que ha pasado el tiempo, se ha avanzado en el cálculo del riesgo de crédito, se explicarán a continuación algunos de los modelos más importantes que hasta no hace mucho eran los que se utilizaban y posteriormente se entrará más en profundidad en los modelos más avanzados. (Mª Valle Carrascal, J., 2015)

1.4.1 BASILEA

- BASILEA I: El modelo comenzó a utilizarse a principios de los años 90, su idea fundamental era que los bancos tuvieran disponible un capital superior al 8% de sus activos en valor nominal según el riesgo que tuvieran los activos, los cuales se clasifican en 4 grupos: 100%, 50%, 20% y 0% de riesgo, de forma que los elementos más seguros como la renta fija tienen una ponderación menor, requiriendo de ese modo, disponer

de un menor capital mínimo en caso de que ese activo sea más seguro. (Comité de Supervisión Bancaria de Basilea, 2006). De esta forma, se establecieron unos principios básicos en los que tenía que estar fundamentada la actividad bancaria.

La parte positiva de este modelo residía en que es muy simple, sin embargo, como defectos tenía varios, que eran principalmente: No tenía en cuenta el vencimiento, el indicador de calificación crediticia, ni si se reducía el riesgo de la cartera por diversificar.

- BASILEA II: Aunque fue aprobado en el año 2004, no se aplicó hasta el 2008. Para determinar la ponderación del riesgo, acogió un sistema mucho más meticuloso, en el que destacaron dos técnicas, la estándar, la cual recurre a la calificación otorgada por una entidad externa, de forma que se distribuye a los deudores en categorías sin tener en cuenta su riesgo de crédito real, y por otro lado, la segunda técnica está basada en la calificación desde dentro, de manera que las propias entidades de financiación determinar el capital mínimo necesario ante el crédito a conceder. Se basa en tres pilares fundamentales:

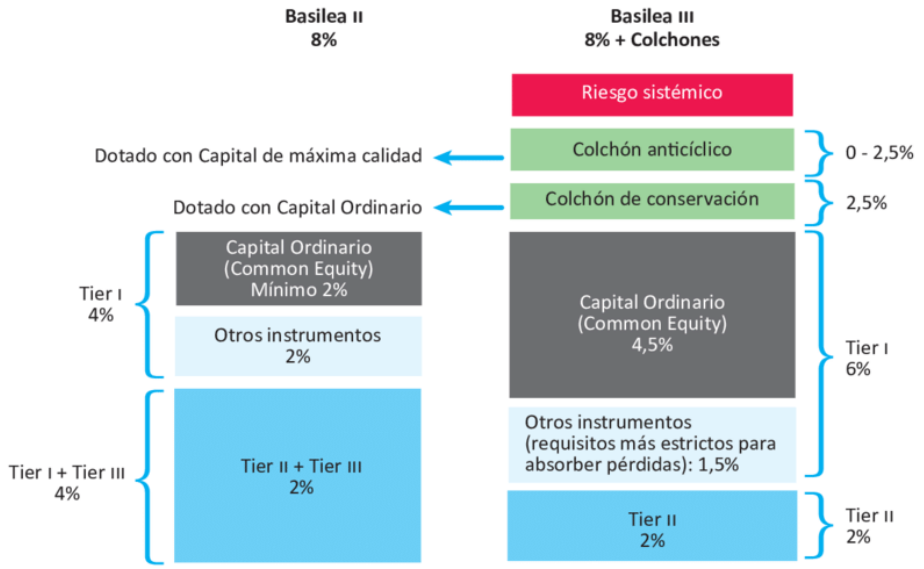
    - Requerimiento mínimo de capital: Relación de capital y riesgo con un mínimo del 8%, ahora se tiene en cuento la calidad crediticia, garantías etc.
    - Supervisión: Coordinación entre bancos y supervisores para garantizar un cumplimiento de la gestión del riesgo.
    - Disciplina de mercado: Divulgar información relevante a inversores, accionistas y mercado sobre los riesgos, situación financiera y similares. Fomentando así la transparencia.

- BASILEA III: Se aprueba en el año 2010. No cambió el método con respecto a Basilea II, pero sí que se introdujeron una serie de medidas que hicieron que aumentara la garantía de que el capital mínimo necesario era el adecuado y suficiente. Tras la crisis, y debido a la gran exposición que tenían los bancos a los "activos tóxicos", se establecieron nuevas medidas como:

    - Endurecer criterios y aumentar la calidad del volumen de capital.
    - Minimizar el riesgo de exposición mediante la actualización de los criterios de cálculo.
    - Establecer un colchón bancario que garantice el resistir un año de crisis.
    - Nuevo ratio de apalancamiento que complementa al ratio de solvencia.

Basilea III se centra en el pánico bancario, y complementa a Basilea I y Basilea II. En la figura 8, se pueden observar las diferencias con respecto a Basilea II.

Figura 8. Basilea II-III.



Fuente: Rubiño-Box, J. A. y Molina-Moreno, V. (2018)

Como se puede observar, cambian los criterios de deuda, requiriéndose en Basilea III los colchones y una mayor cantidad de deuda TIER I en lugar de TIER II y III, es decir, deuda de mayor calidad y cumpliendo los mínimos requeridos de acuerdo con los activos ponderados de riesgo. Los activos ponderados por riesgo (APR) son como el propio nombre indica una multiplicación del valor en libros de los activos por su respectivo factor de riesgo ponderado de acuerdo con los estándares regulatorios y calificación crediticia, de forma que un crédito al consumo tiene un mayor riesgo que una hipoteca, por ejemplo. El cálculo de estos APR permite establecer posteriormente el capital regulatorio mínimo a disponer de la entidad bancaria.

En resumen, los activos ponderados por riesgo se utilizan para calcular los coeficientes de adecuación de capital de los niveles TIER 1, TIER 2 y TIER 3. Estos niveles representan diferentes capas de protección y solvencia dentro de la estructura de capital de una institución financiera, siendo el TIER 1 el nivel más sólido y de mayor calidad.

- BASILEA IV: Existe actualmente una revisión de Basilea III por el Comité de Supervisión Bancaria de Basilea en cuanto a los activos ponderador de riesgo, que podría llamarse Basilea IV pues sería una actualización del actual en la cual se endurecerían los requerimientos de capital para las entidades bancarias. (Contreras, E. 2022).

1.4.2 LAS 5 Cs

Se trata de un modelo muy tradicional que básicamente se basaba en la experiencia y opinión de los que más sabían, de forma que ponían una ponderación a principalmente dos informaciones: La liquidez y fiabilidad que tenía el elemento depositado como aval y, en segundo lugar, algunos ratios acerca de la economía del deudor para ver su solvencia y capacidad de hacer frente al préstamo.

El modelo se llama 5 Cs porque el análisis del modelo estaba desglosado en:

Carácter: Cuál era el expediente de la persona o empresa, su trayectoria económica

Capital: Diferentes ratios que se calculaban para ver su solvencia y capacidad de hacer frentes, endeudamiento…

Capacidad: Cuál era la estabilidad de ingresos del deudor.

Colateral: El elemento depositado como aval, si tenía o no alta liquidez

Ciclo: Cuál es el ciclo económico actual o del sector en el que está el deudor. Tal y como indicaba Taylor, J. (1998), hay una relación entre el ciclo económico de un sector y sus impagos de estos.

1.4.3 CREDIT SCORING

Este modelo se basa en la fórmula [3].

$$Y=2X1 + 4X2 +3X3 + 6X4 + 5X5 + 2X6 + 3X7 \qquad [3]$$

Básicamente hay que saber cuáles son los elementos más importantes acorde a la probabilidad de que no cumpla según los ratios financieros de la persona o empresa, de forma que las variables son:

X2: Estabilidad de las ganancias (Volatilidad del BAII).

X3: Capacidad de servicio de la deuda (BAII/Pagos por intereses).

X4: Solvencia acumulada (Beneficios retenidos/Activos totales).

X5: Liquidez (Activo circulante/Pasivo circulante).

X6: Capitalización (Valor de mercado/Activos totales).

X7: Tamaño (Log (Activos totales)).

Siendo BAII: Beneficios antes de intereses e impuestos.

De forma que si se obtiene un valor de Y menor a uno que se quería obtener se presupone que la probabilidad de impago es alta y no se le concedería el crédito. Este modelo se ha evolucionado también con tendencias similares, siguiendo modelos no lineales.

1.4.4 MODELOS ESTADÍSTICO-FINANCIEROS MÁS ACTUALES

De acuerdo con la tesis de Mª Valle Carrascal, J. (2015), los modelos fueron evolucionando, utilizándose a continuación técnicas más estadísticas:

- Modelos **reducidos**: Destaca el modelo que indica a partir del precio de cotización de un bono cual es la probabilidad de impago. A partir de la calificación que le otorgue una agencia de calificación (Moody's, Fitch o Standard & Poors) a los activos de renta fija, de muy bajo riesgo como los bonos, se calcula una curva de rendimiento para cada calificación de forma que sirva como elemento de valoración ya que se compara con estas curvas acorde al vencimiento del crédito y cupón que se devengue.

- Modelos basados en **datos históricos:** Como su propio nombre indica, se estima el riesgo de crédito a partir de una serie de datos históricos, se basa en el mismo principio que el modelo reducido anterior, pero en este caso está basado en datos del pasado con el objetivo de predecir la tendencia.

- Modelos **estructurales**, el modelo de Merton: Evalúa el riesgo en función de la deuda y capital que dispone la empresa o persona, para evaluar esta deuda y capital, utiliza lo que se denomina "Teoría de opciones" el cual relaciona la renta fija y variable y establece así un diferencial.

1.4.5 MODELOS DE RIESGO CON ALGORÍTMOS DE MACHINE LEARNING PARA SU PREDICCIÓN

Comúnmente se utilizan varios algoritmos de aprendizaje automático. Estimar una probabilidad normalmente supone construir un modelo que sea capaz de predecir con una cierta precisión. (Khemakhem, S. y Younes, B., 2018) Los datos históricos que tienen las entidades de financiación, sirven para construir y ajustar los distintos modelos que se explican a continuación: (López Blanco, L. 2022):

- **Modelo de regresión lineal:** El modelo de regresión lineal es una técnica utilizada para predecir el valor de una variable en función de otra variable. En este modelo, tenemos una variable dependiente que queremos estimar y una o más variables independientes que utilizamos para realizar esa estimación.

  La regresión lineal busca determinar los coeficientes de una ecuación lineal que mejor se ajuste a los datos. El objetivo es encontrar una línea recta o una superficie que minimice las diferencias entre los valores pronosticados por el modelo y los valores reales de la variable dependiente.

  Una vez obtenida la ecuación de regresión lineal, podemos utilizarla para estimar el valor de la variable dependiente a partir de los valores de las variables independientes. Esto nos permite hacer pronósticos o inferir información basada en el modelo ajustado.

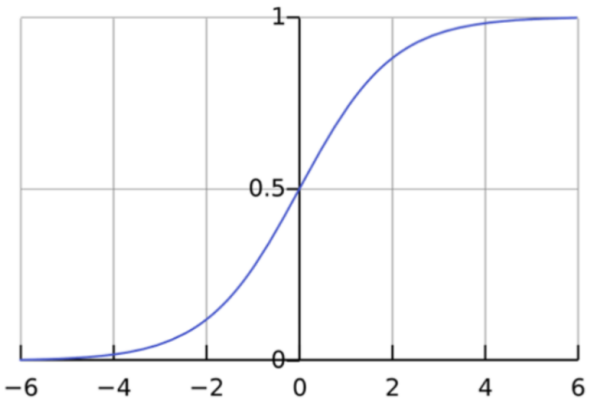
- **Modelo de regresión logística**: Se trata de una función monótona que es creciente entre 0 y 1, la cual, sigue una forma curva que comienza con un crecimiento lento en 0 y va aumentando hasta convertirse en casi exponencial cuando llega a 1. (Martínez Rodríguez, E., 2008)

  Se trata de un modelo de clasificación, por lo que solo sirve para dar un valor binario, o 0 o 1. Este modelo permite predecir pues una variable binaria a partir de una serie variables cuantitativas, sirve para realizar una clasificación de las observaciones en función de que valor tome.

  Se trata de una función también denominada “Logit” que es muy utilizada en el machine Learning y la inteligencia artificial, ya que es poco compleja y a partir de una serie de variables de entrada que se le dan como input y una serie de salidas, outputs, en las que se indica que resultados se obtuvieron. Encuentra una serie de patrones y genera un modelo que es capaz de predecir de nuevo una nueva salida en función de unos datos de entrada que todavía no había visto. Esto se denomina aprendizaje supervisado, ya comentado en el punto anterior.

En la figura 9, se muestra la función *logit*, que devuelve para la variable dependiente entre 0 y 1.

Figura 9. Gráfica regresión logística.



Fuente: Albiol, A. (2022)

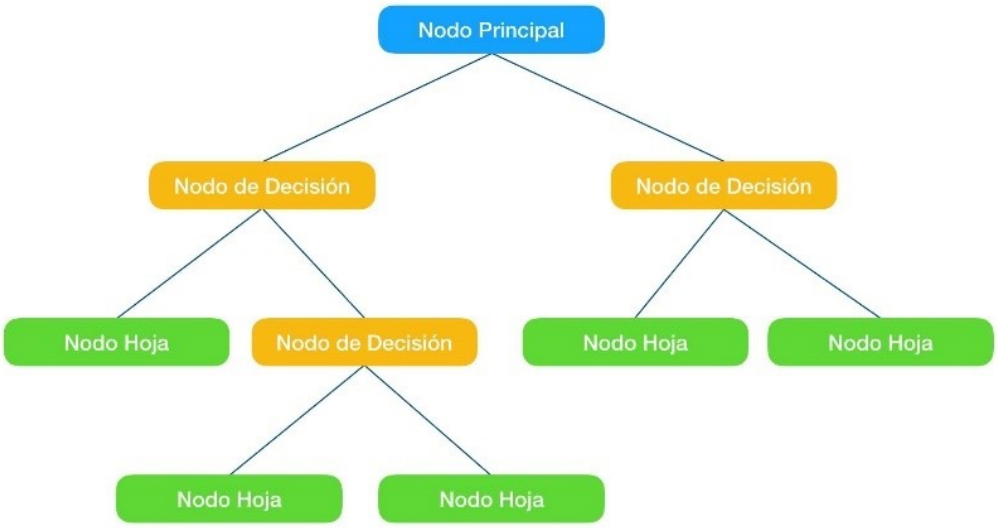
Hay diversos tipos de regresión logística: binaria, en la que solo hay dos resultados posibles, multinomial, donde la variable respuesta puede tomar tres o más variables, por ejemplo, saber si un consumidor prefiere beber cerveza, agua o vino, y por último la ordinal, que es como la anterior, pero siguen un orden las variables, por ejemplo, la preferencia de algo del 1 al 4.

- **Modelos de árboles de decisión:** De acuerdo con JavaTpoint, el árbol de decisiones es una técnica de aprendizaje supervisado que se utiliza tanto en problemas de clasificación como en problemas de regresión, aunque su principal aplicación se encuentra en la clasificación. Este método consiste en un clasificador con una estructura en forma de árbol, donde los nodos internos representan las características del conjunto de datos y las ramas representan las reglas de decisión. Cada nodo hoja del árbol representa un resultado o una clase específica asignada.

  Se distinguen dos tipos de nodos: los nodos de decisión y los nodos hoja. Los nodos de decisión desempeñan el papel de tomar decisiones y presentan múltiples opciones o caminos a seguir, mientras que los nodos hoja representan los resultados o conclusiones de esas decisiones y no tienen más ramificaciones. La toma de decisiones o las pruebas se fundamentan en las características o atributos del conjunto de datos proporcionado. Se llama árbol de decisión porque, similar a un árbol, comienza con el nodo raíz, que se

expande en más ramas y construye una estructura similar a un árbol. En la figura 10, se muestra de forma gráfica el esquema que sigue este modelo:

Figura 10. Árbol decisión.



Fuente: Rodríguez, V. (2018)

- **Modelo KNN o agrupamiento:** Es también una técnica de aprendizaje supervisado, en la que cuando se introduce un nuevo individuo, se clasifica asociándolo al dato más cercano, no intenta predecir a partir de sus datos si no que en función de ellos le otorga el resultado del dato de su memoria que más se aproxime, devolviendo la misma salida que tenía almacenada.

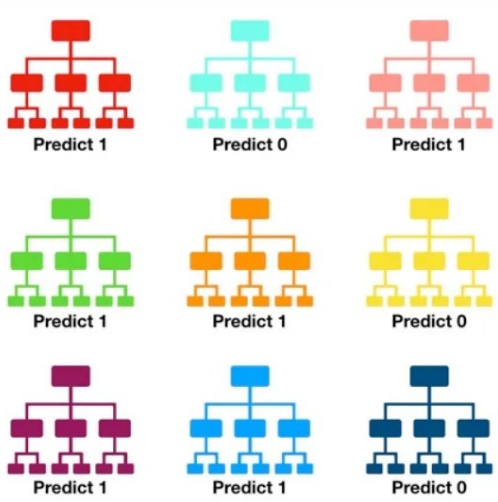
- **Modelo de clasificación de Naïve Bayes**: El clasificador Naïve Bayes es un algoritmo de aprendizaje automático supervisado, que se utiliza para tareas de clasificación, como la clasificación de texto. También es parte de una familia de algoritmos de aprendizaje generativo, lo que significa que busca modelar la distribución de entradas de una clase o categoría determinada. A diferencia de los clasificadores discriminativos, como la regresión logística, no aprende qué características son las más importantes para diferenciar entre clases. La fórmula [4] muestra el teorema.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad [4]$$

Usando el teorema de Bayes, podemos hallar la probabilidad de que suceda A, habiendo sucedido B. Aquí, B es la evidencia y A es la hipótesis. En este caso, se parte de la premisa de que los predictores o características son independientes entre sí. Esto implica que la presencia de una característica en específico no tiene influencia sobre las demás.

- **Modelo "Random forest":** Se trata de un modelo que evolucionado del modelo de los árboles de decisión, (Yiu, T., 2021), *Random forest*, como su nombre lo indica, consiste en una gran cantidad de árboles de decisión individuales que operan como un conjunto. Cada árbol individual en el bosque aleatorio escupe una predicción de clase y la clase con más votos se convierte en la predicción de nuestro modelo tal como muestra la figura 11.

Figura 11. Random Forest.



Fuente: Yiu, T. (2021)

Aunque algunos árboles puedan no ser correctos, otros árboles estarán en lo cierto, por tanto, en conjunto, los árboles se mueven en la dirección correcta, en el caso de la figura anterior, dado que tiene seis 1s y tres 0s, la predicción es 1.

1.4.6 MODELO Z-SCORE DE ALTMAN

Edward I. Altman desarrolló el modelo Z-Score en 1960. Se trata de un modelo que sirve para evaluar el riesgo crediticio de una empresa de forma que es capaz de predecir la posibilidad de que entre en quiebra en el corto plazo. Para ello se basa en la fórmula [5]. (Porté, D., 2020)

$$\text{Altman Z-score} = 1{,}2 * X1 + 1{,}4 * X2 + 3{,}3 * X3 + 0{,}6 * X4 + 1{,}0 * X5 \qquad [5]$$

- X1 = Capital Circulante (WC)/ Activos Totales
- X2 = Beneficios Retenidos/ Activos Totales
- X3 = Beneficio Operativo (EBIT)/ Activos Totales
- X4 = Capitalización Bursátil/ Pasivos Totales
- X5 = Ventas/ Activos Totales

El resultado obtenido al aplicar la fórmula determina la probabilidad de bancarrota de una empresa. Podemos identificar tres escenarios distintos:

1. Escenario seguro: Cuando el resultado es mayor a 2,99, la empresa tiene una salud financiera sólida y no se espera que enfrente dificultades.
2. Escenario ambiguo: Si el resultado se encuentra entre 1,89 y 2,99, la situación es incierta. Existe la posibilidad de que la empresa declare quiebra en los próximos años si no mejora su salud financiera.
3. Escenario peligroso: Cuando el resultado es menor a 1,89, la empresa enfrenta un alto riesgo de quiebra en el corto o mediano plazo.

En los últimos años, se ha observado que un Z-Score más próximo a 0 sugiere que una empresa puede estar enfrentando dificultades financieras. Durante una conferencia realizada en 2019 llamada "50 años de la puntuación de Altman", el profesor Altman en persona mencionó que los datos más recientes han revelado que el valor crítico a tener en cuenta para evaluar la solidez financiera de una empresa es 0, no 1,8, como se pensaba anteriormente (Kenton, W., 2022).

El Z-Score y el modelo de regresión logística son enfoques distintos para analizar el riesgo crediticio. El Z-Score se basa en ratios financieros ponderados para evaluar la salud financiera de una empresa y predecir la probabilidad de bancarrota. Por otro lado, la regresión logística utiliza variables independientes para predecir una categoría binaria, como la “quiebra” o “no quiebra”. Mientras que el Z-Score proporciona un valor numérico, la regresión logística estima coeficientes que indican la influencia de cada variable, y es la regresión logística, el modelo que se utilizará en este modelo.

CAPÍTULO 2: DESARROLLO E IMPLEMENTACIÓN DEL MODELO DE RIESGO DE CRÉDITO

2.1 SELECCIÓN DE LA TÉCNICA DE MODELADO ELEGIDA

Para realizar este modelo se ha optado por utilizar la regresión logística como técnica para elaborar el modelo. Como se ha explicado previamente, la regresión logística es un modelo sencillo para responder a la pregunta de nuestro modelo: se concede el crédito o no se concede el crédito, es pues una variable respuesta binaria, 0 o 1.

2.2 PREMISAS DEL MODELO

El modelo que se ha desarrollado data del año 2011, por lo que la similitud de los resultados no debe compararse con lo que se pueda encontrar a fecha de hoy. Asimismo, proviene de una base de datos de carácter pública que, a pesar de haber sido utilizada para una competición, los datos con los que se trabajan pueden no ser 100% reales o provenir de entidades de financiación no importantes o fiables. Además, es una base de datos de EEUU por lo que se ha de tener en cuenta que allí los requisitos de crédito pueden diferir de los de España.

La base de datos con la que ha entrenado el modelo dispone de más de cien mil muestras por lo que se presupone que se representa la totalidad de la población.

2.3 ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN Y PLATAFORMA DE DESARROLLO

Como lenguaje de programación, se ha elegido usar Python. Python es un lenguaje de programación de nivel alto interpretado, caracterizado por ser orientado a objetos y contar con semántica dinámica. Sus estructuras de datos integradas, de gran nivel, junto con su capacidad de escritura y enlace dinámicos, lo convierten en una opción muy atractiva para el desarrollo rápido de aplicaciones. Python pone un fuerte énfasis en la legibilidad del código, lo que a su vez reduce los costos asociados al mantenimiento del programa. Además, Python ofrece soporte para la utilización de módulos y paquetes, fomentando así la modularidad y la reutilización de código. Dispone también de una extensa biblioteca para descargar de manera gratuita de multitud de herramientas, como los distintos modelos de Machine Learning y en nuestro caso el de regresión logística.

En cuanto a la plataforma para el desarrollo del programa, se comenzó utilizando Google Colab, un entorno de programación de Google con lenguaje Python, ya que permite trabajar en línea,

archivando los datos en Google Drive. La segunda parte positiva que tiene es que se ejecuta en los servidores de Google, de forma que no solo no consume a nivel computacional sino que además está siempre actualizado con los distintos paquetes y librerías.

Posteriormente, una vez el modelo estaba desarrollado, se pasó a utilizar Visual Studio Code, una plataforma multilenguaje de programación que se utiliza también en algunas asignaturas de la universidad, en este, se descarga el paquete de Python y se trabaja de la misma forma. El motivo de proseguir el código en un entorno local fue por la simplicidad para generar un Dashboard, un "Dash" es un lugar donde se muestran todos los datos que se quieran mostrar en un único lugar, como si fuera una pequeña página web que se ejecuta en un servidor local, es básicamente el lugar donde posteriormente se introducirán los datos, se comunicará con el modelo y responderá si se le concede o no el crédito.

2.4 DESCRIPCIÓN DETALLADA DE LA IMPLEMENTACIÓN DEL MODELO

En primer lugar, se procede a descargar la base de datos de la plataforma Kaggle, la base de datos se denomina "Give me some credit" y se compone de una serie de archivos:

1. Data Dictionary.xls: Archivo que contiene el nombre de las variables que figuran en la base de datos.
2. Cs-training.csv: Es el archivo que contiene todo el conjunto de datos que servirán para entrenar al modelo.
3. Cs-test.csv: Con este archivo se comprobará la precisión del modelo entrenado con training comparándolo con este, de forma que se sabrá cuantos ha acertado y cuantos ha fallado.
4. SampleEntry: Este archivo es una muestra con probabilidades, el cual no utilizaremos.

Una vez descargada la base de datos, se procede a elaborar el código, antes de comenzar, en la tabla 1, se muestra el nombre de las distintas variables que componen el "dataset" o conjunto de datos:

Tabla 1. Variables del modelo.

Nombre de la variable	Descripción	Tipo
SeriousDlqin2yrs	**Variable respuesta**, persona que ha experimentado hace poco la morosidad de pago	Si/No
RevolvingUtilizationOfUnsecuredLines	Balance de líneas, créditos, hipotecas… dividido entre la suma total de créditos	Porcentaje
age	Edad de la persona	Entero
NumberOfTime30-59DaysPastDueNotWorse	Representa el número de veces que una persona ha estado atrasada en un pago de 30 a 59 días, pero aún no ha experimentado una morosidad más grave.	Entero
DebtRatio	Pagos de deuda y vida divididos entre los ingresos totales	Porcentaje
MonthlyIncome	Ingresos mensuales	Real
NumberOfOpenCreditLinesAndLoans	Número de créditos y líneas abiertas como hipotecas, prestamos de coche…	Entero
NumberOfTimes90DaysLate	Veces que se ha retrasado 90 días o más en su pago	Entero
NumberRealEstateLoansOrLines	Número de hipotecas y préstamos incluyendo líneas de crédito del hogar	Entero
NumberOfTime60-89DaysPastDueNotWorse	Representa el número de veces que una persona ha estado atrasada en un pago de 60 a 89 días, pero aún no ha experimentado una morosidad más grave.	Entero
NumberOfDependents	Número de personas que dependen económicamente de la persona que solicita el crédito	Entero

Fuente: Elaboración propia

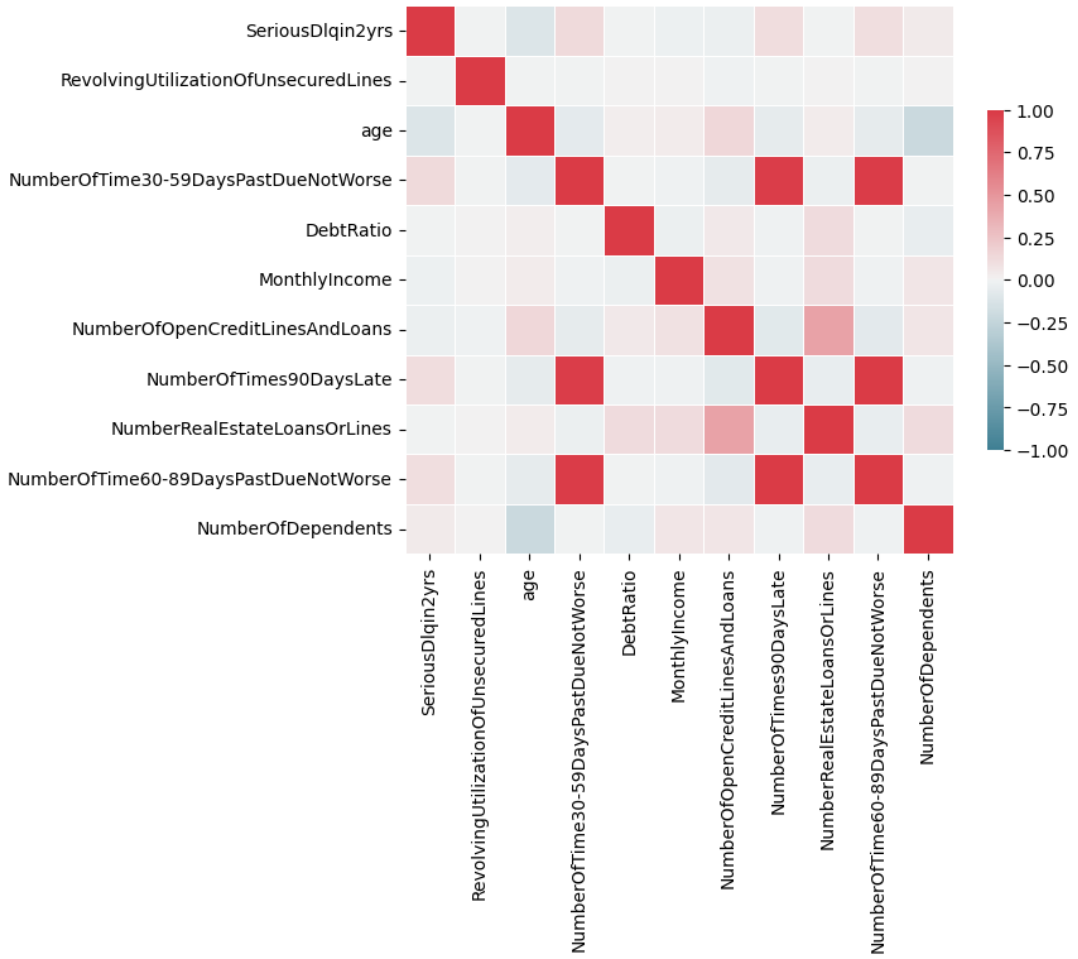
A continuación, se procederá a explicar el código realizado de forma esquemática:

En primer lugar, tal y como se puede ver en el anexo, se han importado las distintas librerías que permiten realizar una serie de funciones como pandas (leer archivos csv), numpy (para operaciones matemáticas, o sklearn.metrics (para la regresión logística).

Una vez importadas las librerías, se han eliminado aquellas filas que tuvieran valores “NaN” con el fin de que aquellas que no tienen información no influyan en el modelo. Se ha calculado también el porcentaje de datos que son de entrenar y los que son de test, quedando en un 40% de test y un 60% de entrenamiento.

Una vez hecho esto, se ha dibujado una matriz de correlaciones, la cual nos permite ver el grado de correlación de las variables y poder así prescindir de campos o variables que están explicadas por otras, para ello se decide visualizarla mediante un mapa de calor, obteniendo la figura 12 :

Figura 12. Mapa de calor.



Fuente: Elaboración propia

Como se observa en el mapa, el cual como indica la leyenda aquellas que tienen un grado de color más rojo tienen una mayor correlación, algunas de las variables tienen un grado de correlación de 1 (a parte de la diagonal por ser ellas mismas con ellas mismas), por lo que se ha procedido a eliminar aquellas en las que sucede esto para evitar problemas de colinealidad. Las variables totalmente correladas son pues:

- NumberOfTimes90DaysLate con NumberOfTIme30-59DaysPastDueNotWorse
- NumberOfTimes90DaysLate con NumberOfTIme60-89DaysPastDueNotWorse

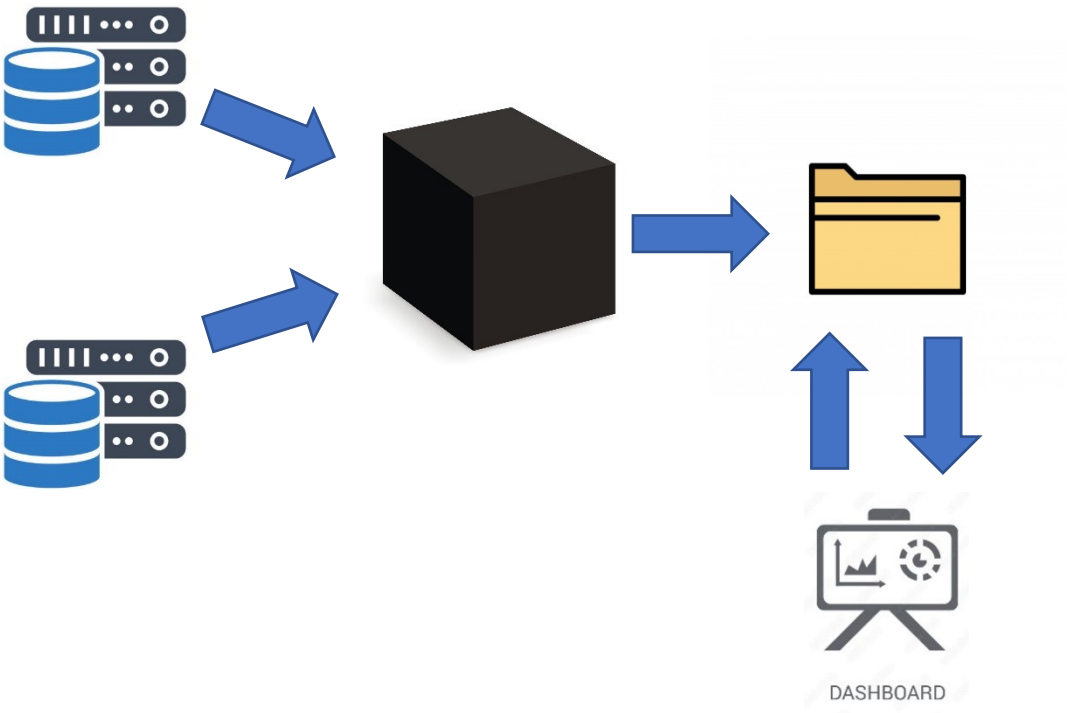
Decidiendo eliminar: NumberOfTIme30-59DaysPastDueNotWorse y NumberOfTIme60-89DaysPastDueNotWorse.

Una vez ya se dispone de las variables adecuadas, se procede a entrenar el modelo, antes de hacerlo, se ha eliminado la primera columna de los datos del train, que es la variable respuesta, y se ha hecho lo mismo con el del test. Después se ha procedido a realizar una estandarización de los datos y se ha realizado la regresión logística, que gracias a un comando "accuracy" podemos saber cuál es la precisión obtenida del modelo que ha generado en la columna a parte del test con respecto a la columna que se había separado del test. Se ha obtenido una precisión del 71,27%, es decir, que de cada 100 filas de inputs ha acertado 71 outputs.

Por último, en la elaboración del modelo, se ha exportado el modelo y el "escaler", archivo que permitirá que cada variable nueva de entrada al modelo tenga la misma contribución que las anteriores y esté en el mismo formato. Estos dos archivos se utilizarán en la elaboración del Dashboard.

A continuación, se muestra la figura 13, que explica el funcionamiento de las distintas partes que componen la elaboración del modelo:

Figura 13. Esquema modelo.



Fuente: Elaboración propia

Como se observa en la figura, el TFG consta de 4 fases independientes:

1. Las bases de datos de donde se extrae toda la información que utilizará el modelo, en este caso son los ficheros CSV de “Give me some credit”.
2. La “caja negra”, la cual es el modelo, como se ha explicado previamente, en el interior se crea una red neuronal que asigna una serie de ponderaciones a las distintas variables, cada una formando una neurona, siendo capaz de identificar los patrones. En este trabajo se ha realizado utilizando una regresión logística.
3. Esta caja negra genera un archivo, que es el modelo ya entrenado, llamado “finalized_model.sav”, el cual permite actuar como el “cerebro”.
4. El Dashboard que se explicará a continuación, que es una forma de visualizar e interactuar con el modelo generado, el doble sentido de las flechas nos indica la existencia de una comunicación bilateral de las partes.

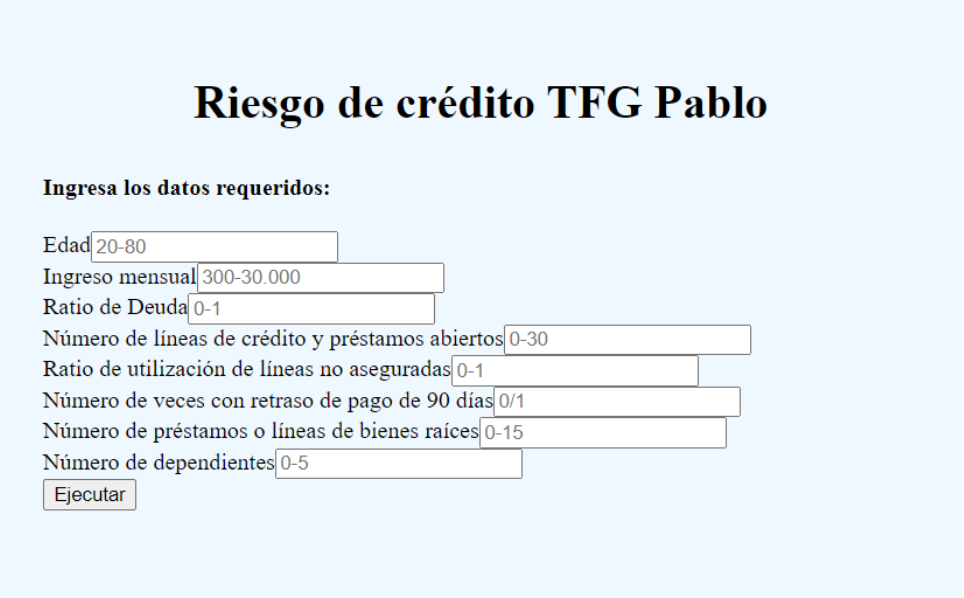
Para la elaboración del Dashboard se ha realizado en un archivo Python distinto, el cual se puede ver en la segunda parte del anexo. A continuación, se detalla por pasos como se ha realizado:

1. Se importan las bibliotecas necesarias, incluyendo Dash, pandas, pickle y sklearn.
2. Se cargan el modelo y el escalador (scaler) previamente entrenados a partir de archivos guardados.
3. Se crea una instancia de la aplicación Dash.
4. Se define el diseño de la aplicación utilizando elementos HTML y componentes Dash, como **html.Div**, **html.H1**, **dcc.Input**, **html.Button**, etc.
5. Se define una función de devolución de llamada (callback function) que se activa cuando se hace clic en el botón "Ejecutar".
6. En la función de devolución de llamada, se capturan los valores ingresados por el usuario y se realiza una transformación de los datos.
7. Se utiliza el modelo cargado para hacer una predicción basada en los datos ingresados.
8. Se muestra el resultado de la predicción en la interfaz de usuario, con un mensaje indicando si se ha concedido o no el crédito.
9. Si ocurre algún error durante el procesamiento de los datos, se muestra un mensaje de error correspondiente.
10. La última línea **if __name__ == '__main__':** garantiza que la aplicación se ejecute solo cuando se ejecute este archivo directamente, no cuando se importe como un módulo.

En resumen, este código utiliza Dash para crear una interfaz web interactiva que permite a los usuarios ingresar información financiera y obtener una evaluación de riesgo crediticio en función de un modelo de aprendizaje automático previamente entrenado. El diseño de la interfaz y la lógica de procesamiento se definen utilizando componentes Dash y funciones de devolución de llamada.

En la figura 14, se muestra una captura de pantalla antes de introducir los datos en el Dashboard y después de hacerlo.

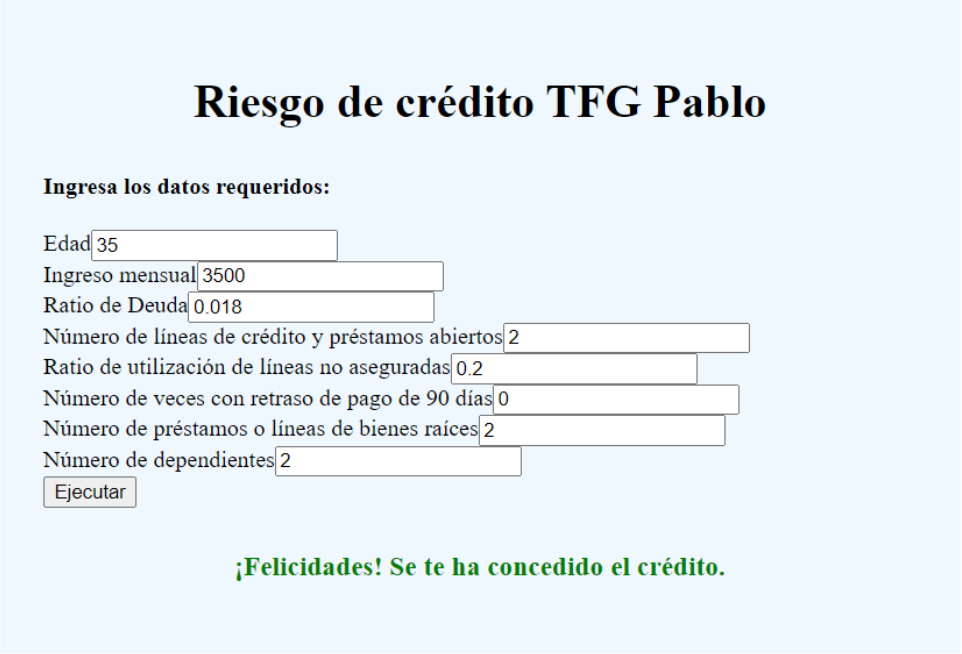
Figura 14. Dashboard sin datos.



Fuente: Elaboración propia

En la figura 15, se muestra una vez rellenados los campos y ejecutado para unas condiciones favorables.

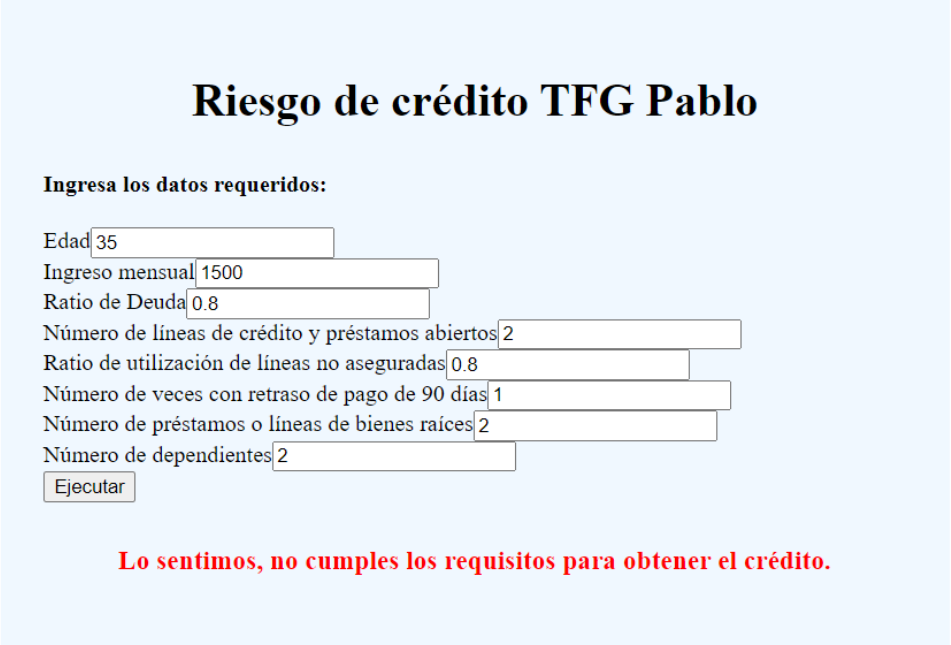
Figura 15.Dashboard con datos.



Fuente: Elaboración propia

Como se puede observar, el modelo predice que se le puede dar un crédito para esos requisitos. Sin embargo, si variamos los parámetros, incrementando las variables que perjudican, obtenemos la respuesta de que no se concede el crédito como se muestra en la figura 16.

Figura 16.Dashboard crédito no concedido.



Fuente: Elaboración propia

CAPÍTULO 3: ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

3.1 EVALUACIÓN Y REFLEXIÓN

En primer lugar y para realizar un mejor análisis del modelo, se han añadido unas líneas de código, las cuales permiten extraer el valor de los coeficientes que han sido asignados a cada variable.

Coeficientes de ponderación de las variables que intervienen:

- RevolvingUtilizationOfUnsecuredLines: -0.0253
- DebtRatio: -0.0325
- Age: -0.3976
- MonthlyIncome: -0.4056
- NumberRealEstateLoansOrLines: 0.0881
- NumberOfOpenCreditLinesAndLoans: 0.0917
- NumberOfDependents: 0.1056
- NumberOfTimes90DaysLate: 3.1809

Se han colocado en orden de importancia, y se interpretan de la siguiente manera:

Un coeficiente negativo indica que tiene una relación inversa con la probabilidad de que devuelva el crédito, es decir, un aumento en el valor de la variable se relaciona con que disminuye la probabilidad de impago y viceversa.

Por otro lado, la magnitud del coeficiente indica cuál es la importancia que tiene esa variable para determinar la probabilidad del impago, por lo que la variable “NumberOfTimes90DaysLate” (número de veces que se ha retrasado en el pago 90 o más días), tiene un gran impacto para determinar el resultado. Asimismo, la variable que más importa si alguna ha de aumentar es “MonthlyIncome” (ingresos mensuales).

Si analizamos las ponderaciones observamos un razonamiento coherente, las variables “RevolvingUtilizationOfUnsecuredLines”, “DebtRatio”, “NumberRealEstateLoansOrLines”, “NumberOfOpenCreditLinesAndLoans” y “NumberOfDependents “ tienen un valor casi despreciable, importando principalmente “age”, “MonthlyIncome” y “NumberOfTimes90DaysLate” y el signo positivo y negativo corroboran la lógica, ya que a medida que aumenta la edad se le concede antes el crédito, hasta cierto limite, ya que las personas mayores tienen un historial crediticio más estable, y por otra parte, a medida que los ingresos aumentan también disminuye la probabilidad de impago.

Por otro lado, tiene un gran peso la variable "NumberOfTimes90DaysLate" de forma que, si esta aumenta, la probabilidad de impago aumenta mucho, tiene sentido que esto sea así, ya que el registro de morosidad es algo que tiene muy en cuenta por las entidades de financiación a la hora de conceder un crédito.

Se han realizado numerosas pruebas en el Dashboard y se ha observado, teniendo en cuenta que su porcentaje de aciertos es de un 71,27%, es que predice con precisión aquellos valores introducidos que sean coherentes, es decir, el modelo ha establecido unos patrones de forma que si se le presentan casos relativamente conocidos funciona de una manera correcta, sin embargo, si se le introducen datos que no está acostumbrado a ver, aunque tenga asignadas unas ponderaciones, el modelo no funciona. Esto sucede porque el modelo no es capaz de asociarlo a algo que ya ha visto por ser muy diferente, de modo que se produce un error y devuelve un resultado no coherente, una forma de solucionar esto sería mejorando el código de forma que se construya un modelo matemático que tenga un ajuste más adecuado, para ello se podría haber realizado por ejemplo "Random forest", pero este habría aumentado la complejidad del código y no era el objetivo principal del TFG.

En la tabla 2 figuran una serie de estadísticos que se han extraído, que nos permiten conocer más en profundidad los valores y funcionamiento del modelo.

Tabla 2. Modelo logit obtenido

VARIABLE	Coeficientes del modelo	|t|	P-Value
RevolvingUtilizationOfUnsecuredLines	-0.0253	0.8	0.42
Edad	-0.3976	24.9	1.72e-137
DebtRatio	-0.0325	2.6	8.96e-03
MonthlyIncome	-0.4056	11	1.55e-28
NumberOfOpenCreditLinesAndLoans	0.0917	0.6	0.55
NumberOfTimes90DaysLate	3.1809	14.7	5.11e-49
NumberRealEstateLoansOrLines	0.0881	5.4	5.19e-08
NumberOfDependents	0.1056	11.4	3.03e-30
R^2 AJUSTADA		71.26%	

Fuente: Elaboración propia

Tal y como se observa en la tabla, la bondad del ajuste del modelo (la precisión) es el del 71,26%, esto quiere decir que las variables de entrada son capaces de explicar el 71,26% del resultado del modelo. Por otro lado, se han extraído la media, desviación típica, valor máximo y valor mínimo de cada una de las variables, de forma que esto ayudará a posteriormente indicar al usuario un rango de utilización del modelo para no introducir valores fuera de lo ya visualizado por el modelo.

Es importante destacar, que también se ha extraído el test de significación |t|, en valor absoluto para cada una de las variables, donde se ha obtenido que hay dos variables que realmente no influyen en el modelo por estar por debajo de dos, éstas son: "RevolvingUtilizationOfUnsecuredLines" y "NumberOfOpenCreditLinesAndLoans". Aunque si se quitaran no variaría el resultado y comportamiento del modelo, se ha decidido dejarlas porque esto permite también obtener una mayor información del solicitante, y además, dado que en la base de datos se incluían, resulta así más sencilla su elaboración.

Por otro lado, se ha obtenido que el p-value para las distintas variables es también para estas variables no significativas pues su valor es 0,42 y 0,55, siendo muy superiores a 0,05 o 0,1 que nos daría un intervalo de confianza adecuado. En el caso del resto de las variables si se han obtenido valores muy inferiores a 0,05.

MANUAL DE FUNCIONAMIENTO

Para el uso del programa hay que ejecutar en primer lugar el archivo "TFG" que genera a partir de la base de datos, el modelo que se usará en el archivo "dashboard" que posteriormente hay que ejecutar, tras ejecutarlo, aparecerá una dirección en local que te lleva a una web en la que hay que introducir los parámetros correspondientes a las variables indicadas en el apartado anterior. Una vez introducidos todos los datos, se ejecuta el modelo de forma automática, indicando el mensaje respuesta que se desea obtener. A continuación, se explica de manera detallada los pasos a realizar, así como un diccionario/ayuda de las variables:

1. Ejecutar archivo "TFG" desde Python
2. Ejecutar archivo "dashboard" desde Python
3. Clicar en la dirección IP que aparece en la ventana de Python:

Figura 17.Captura del dashboard donde se indica la dirección.

Fuente: Elaboración propia

Como se puede observar en la figura 17, aparece una dirección: http://127.0.0.1:8050/ en la cual hay que clicar y nos llevará a una web de nuestro navegador.

4. Rellene los campos que aparecen en el navegador que visualiza de acuerdo con la información de la tabla 3.

Tabla 3. Diccionario con instrucciones.

VARIABLE	SIGNIFICADO	RANGO RECOMENDADO	ORDEN IMPORTANCIA
RevolvingUtilizationOfUnsecuredLines	Balance de líneas/créditos, dividido entre créditos	0-1	7
Edad	Edad	20-80	3
DebtRatio	Ratio de deuda	0-1	6
MonthlyIncome	Ingresos mensuales	300-30.000	2
NumberOfOpenCreditLinesAndLoans	Número de líneas de crédito abiertas	0-30	5
NumberOfTimes90DaysLate	Número de veces de demora en pago de deuda	0	1
NumberRealEstateLoansOrLines	Número de créditos de hipotecas	0-15	6
NumberOfDependents	Número de personas que dependen de su economía	0-5	4

Fuente: Elaboración propia

5. Una vez rellenado, se ejecuta tras presionar el botón de “ejecutar”, de forma instantánea se proporciona el mensaje:
    a. ¡Felicidades! Se te ha concedido el crédito
    b. Lo sentimos, no cumples los requisitos para obtener el crédito

    Se recomienda independientemente del mensaje obtenido, consultar el siguiente punto del TFG en el que se incluyen algunos consejos para mejorar condiciones e incluso pasar del mensaje “b” al “a”.

6. Revise por un experto la operación para confirmar el acierto del modelo.

3.2 CONSEJOS PARA EL CLIENTE QUE SOLICITE EL CRÉDITO

Como se ha podido visualizar en las pruebas de funcionamiento y en apartados anteriores, existen una serie de variables que tienen un importante peso en la determinación de la concesión del crédito. El objetivo de este apartado reside en determinar consejos o propuestas de mejora para aquellos clientes que no han sido autorizados para su obtención, o aquellos que sí que han obtenido la aprobación del modelo, pero quieren asegurarse de esto, además de obtener unas mejores condiciones una vez se conceda el crédito, ya que, como es obvio, alguien que esté en el límite y si se le conceda el crédito, no dispondrá de las mismas condiciones de aquel cliente que sea el ideal, sin deuda y altos ingresos etc.

En primer lugar, los ingresos mensuales son la variable más importante que se tiene en cuenta cuando se concede el crédito, por ello, se recomienda al cliente que dé a conocer a la entidad bancaria, no solo los ingresos de la nómina, sino que además se incluya todo ingreso del que disponga, ya sea renta de un piso, dividendos de acciones, ayudas económicas etc. La siguiente variable que más ayuda es la edad, sin embargo, esta variable no se puede mejorar por ser algo que no está sujeto a modificación.

Por otro lado, también es importante prestar atención a aquellas variables que perjudican a la concesión en el caso de que éstas aumenten, como por ejemplo y de forma principal, ser perteneciente a la lista de morosos (ASNEF, CIRBE y RAI) o estar en vigilancia especial. Es importante que en el historial crediticio del cliente no figuren retrasos en los pagos de deudas, ya que tienen un importante peso o ponderación en el cálculo de determinación de la concesión del crédito.

3.3 OTRAS APLICACIONES DEL MODELO

El modelo elaborado puede servir para otras aplicaciones similares, con una ligera modificación de la base de datos o variables puede servir para evaluar a las empresas, evaluar el riesgo de proveedores, evaluar riesgos de una inversión, evaluar avales para empresas, detectar fraudes por el hecho de realizar actividades fraudulentas de carácter sospechoso, y por supuesto para evaluar el riesgo en préstamos que son no bancarios, por parte de otras entidades o personas físicas.

Para poder realizar este tipo de modelos, se necesita como hemos hecho en este TFG, de una gran base de datos y a continuación realizar el mismo proceso que se ha hecho en el TFG, elaborar un modelo matemático que genere un archivo CSV que pueda posteriormente comunicarse con el dashboard para determinar la variable respuesta.

En resumen, dado que el modelo utiliza la regresión logística, realizando una modificación de la base de datos y adaptando el código, se podrán realizar aquellos modelos cuya variable respuesta sea binaria, ya que la regresión logística es una función que devuelve dos posibles valores.

3.4 VENTAJAS Y ASPECTOS IMPORTANTES ACERCA DEL MODELO

Tal y como se recalca en los objetivos del TFG, a parte de la parte fundamental que es el comprender y saber manejar el riesgo de crédito, está el de elaborar el modelo de riesgo que pueda ayudar a una entidad de financiación a determinar la decisión de conceder un crédito.

El hecho de disponer de una herramienta que procese miles de datos históricos sobre clientes similares, determinando así si se debe de conceder o no el crédito, de acuerdo con el histórico de aquellos que han cumplido el pago de su deuda y los que no, es sin duda una herramienta de gran ayuda que, de hecho, toda entidad financiera dispone hoy en día.

El modelo matemático ayuda a incrementar la precisión de la decisión que ha de tomar la entidad ya que el modelo siempre va a ser objetivo, es además muy flexible, siendo posible ajustarlo y adaptarlo conforme a las necesidades de cada institución financiera.

Es importante destacar que el coste de implementación del modelo para la entidad de financiación es muy bajo, ya que se trata de un código de Python que ha sido elaborado con librerías públicas y gratuitas y solamente se ha de disponer de una base de datos extensa, un ordenador capaz de procesar y ejecutar el modelo, y una persona con conocimientos de

programación capaz de adaptar el modelo a las necesidades de la institución, así como mejora de éste en caso requerido.

Por otro lado, se ha de tener en cuenta, que, dada la complejidad del modelo y su carácter académico, el modelo tiene como público para que lo utilicen las entidades de financiación de tamaño reducido, en especial para las Fintech ya que éstas, suelen optar por entornos digitales incrementando así la eficiencia y costes. En cuanto al público para el que ha sido enfocado el modelo es para clientes minoristas, más concretamente, la base de datos son créditos al consumo mayoritariamente, por lo que, si se desea otro fin, se ha de modificar ligeramente el modelo y base de datos de acuerdo con las necesidades de cada usuario. El modelo se podrá utilizar siempre que sea necesario estimar la calidad crediticia.

CAPÍTULO 4: CONCLUSIONES Y MEJORAS

Como conclusión al TFG realizado, se comenzará recalcando que se han logrado los objetivos iniciales de elaborar un modelo que sea capaz de decirle a un solicitante de un crédito si se le concede o no el crédito. Se han obtenido resultados con una precisión del 71,27% en comparación a los datos de "train" por lo que se trata de un modelo bastante fiable teniendo en cuenta su ámbito académico.

El trabajo desarrollado en el TFG además de ser muy interesante es de una tremenda utilidad, ya que no solo se ha desarrollado un producto sencillo que se asemeja a cómo funcionan en la realidad estos modelos de los bancos, sino que además, ha servido para conocer en profundidad cómo funcionan tanto los modelos predictivos de carácter básico de Machine Learning, como los datos y ponderaciones a los que se les otorga más importancia a la hora de conceder un crédito, pudiendo comprender dos aspectos fundamentales en cuanto al riesgo de crédito de la actualidad.

Para responder a la utilidad, se debe recordar que la base de datos del modelo proviene de Kaggle, una base de datos de carácter pública del año 2011 y nacionalidad americana, por lo que los resultados obtenidos provienen de créditos ya concedidos por algoritmos y analistas de dicha entidad de financiación de la cuál provienen los datos. Asimismo, cabe recalcar que el objetivo del presente TFG no era únicamente elaborar un programa capaz de acertar el 100% de las veces para venderlo a una entidad bancaria, si no que al tratarse de un trabajo de empresa y no de programación, se trataba de comprender bien cómo funcionan los modelos de riesgo de créditos actuales, como se elaboran, como manejar el riesgo, y cuáles son los parámetros que se tienen en cuenta.

Para mejorar el modelo realizado en este TFG, se recomendaría realizar las siguientes tareas:

- Obtener una base de datos actualizada a 2023 de origen nacional y de una entidad bancaria de forma directa.
- Realizar un código que sea computacionalmente más eficiente.
- Desarrollar un modelo más sofisticado que obtenga una precisión de aciertos superior. Es decir, utilizar modelos matemáticos más complejos y precisos que la regresión logística.
- En lugar de realizar una pantalla como Dashboard, desarrollar una aplicación ejecutable en servidores públicos accesible para todo el mundo. Además de darle un aspecto más profesional y visual.
- Disponer de un modelo el cual se puedan elegir las variables del modelo, de forma que se puedan introducir únicamente unos parámetros que se deseen, bien sean los

utilizados u otros que se quiera utilizar, por lo que habría que disponer de un modelo distinto que pueda responder a algo que no haya visto, para ello habría que entrar en aprendizaje no supervisado.

- Automatizar la aplicación en tiempo real, de forma que se lleve una planificación que permita actualizar la base de datos en tiempo real, adaptándose al momento temporal en el que se ejecuta. Esto permitiría adaptarse al momento económico, por ejemplo, en el caso de que se encuentre ante una crisis como la de la pandemia del 2019, adaptar el modelo de forma rápida a la situación económica.

BIBLIOGRAFÍA

1) Calder (1999) *Financing the American Dream*, 118, 124–135. The estimate comes from Arthur H. Ham, "Remedial Loans as Factors in Family Rehabilitation," *Proceedings of the National Conference of Charities and Correction* (New York: Russell Sage Foundation, Department of Remedial Loans, 1911), p. 11.

2) Calvo, D. (2019). *Aprendizaje supervisado* < https://www.diegocalvo.es/aprendizaje-supervisado/> [Consulta: 11 de abril de 2023]

3) Calzada, A. (2020). *Todo lo que debes saber sobre el riesgo crediticio.* <https://www.advancing.es/blog/inversion/calcular-riesgo-crediticio/> [Consulta: 7 de abril de 2023]

4) España. Circular 4/2016, del 27 de abril, del Banco de España sobre normas de información financiera pública y reservada y modelos de estados financieros. *BOE*, 6 de mayo de 2016, núm. 110, p. 30319-30422.

5) Comité de Supervisión Bancaria de Basilea (2006). "Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework–Comprehensive Version". *Banco de Pagos Internacionales.*

6) Contreras, E. (2022). *Basilea IV colma el vaso de las empresas al encarecer el crédito* <https://www.eleconomista.es/empresas-finanzas/noticias/11717995/04/22/Basilea-IV-colma-el-vaso-de-las-empresas-al-encarecer-el-credito.html> [Consulta: 25 de mayo de 2023]

7) Fernández, L. (2023*). Criterios que sigue el banco para conceder o rechazar un préstamo*. < https://www.rankia.com/blog/mejores-creditos-y-prestamos/3527566-que-criterios-sigue-banco-para-concederte-rechazarte-prestamo> [Consulta: 11 de abril de 2023]

8) Gutiérrez, I. (2022). *Cuál es la historia del crédito y del sistema bancario.* <https://muyfinanciero.com/historia/historia-del-credito/> [Consulta: 1 de abril de 2023]

9) Jagtiani, J., Lemieux, C. (2019). "Do fintech lenders penetrate areas that are underserved by traditional banks?" en *Journal of Economic Behaviour and Organization*, 100, pp. 43-54.

10) JavaTpoint. *Decision Tree Classification Algorithm*. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm> [Consulta: 10 de abril de 2023]

11) Kavlakoglu, E. (2020). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?*. <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks> [Consulta: 11 de abril de 2023]

12) Kenton, W. (2022). *Altman Z-Score: What It Is, Formula, How to Interpret Results.* <https://www.investopedia.com/terms/a/altman.asp> [Consulta: 5 de junio de 2023]

13) Khemakhem, S., Younes, B. (2018) "Predicting credit risk on the basis of financial and non-financial variables and data mining." *Review of Accounting and Finance* 17.3: 316-340.

14) Kielar, H. (2021). *The History of Credit*. <https://www.rocketmoney.com/learn/debt-and-credit/the-history-of-credit> [Consulta: 3 de abril de 2023]

15) López Blanco, L. (2022*). Explicación y predicción del default en créditos, con la implementación de modelos de Machine Learning*. Tesis. Madrid: Universidad de Comillas.

16) Mª Valle Carrascal, J. (2015). *Modelos de medición del riesgo de crédito*. Tesis. Madrid: Universidad Complutense de Madrid.

17) Martínez Rodríguez, E. (2008). *Logit Modelo como modelo de elección discreta: origen y evolución*. Universidad Complutense de Madrid. <Documat-LogitModelComoModeloDeEleccionDiscreta-2652092> [Consulta: 9 de abril de 2023]

18) Olegario, R. (2019). *The History of Credit in America*. <https://oxfordre.com/americanhistory/display/10.1093/acrefore/9780199329175.001.0001/acrefore-9780199329175-e-625;jsessionid=51E72339489512E8825C90AD3B6FC248#acrefore-9780199329175-e-625-note-22> [Consulta: 1 de abril de 2023]

19) Peterdy, K. (2023). *Credit Risk. The risk that a borrower may not repay credit obligations* <https://corporatefinanceinstitute.com/resources/commercial-lending/credit-risk/> [Consulta: 8 de Abril de 2023]

20) Porté, D. (2020). *Altman Z-Score: la fórmula para detectar empresas en riesgo de quiebra.* < https://www.bolsaexpertos.com/altman-z-score/> [Consulta: 5 de junio de 2023]

21) Rodríguez, V. (2018). *Decision trees / Árboles de decisión para clasificar en Python*. <https://vincentblog.xyz/posts/decision-trees-arboles-de-decision-para-clasificar-en-python> [Consulta: 10 de abril de 2023]

22) Rubiño-Box, J. A. y Molina-Moreno, V. (2018). *Retos del sector financiero cooperativo español ante el riesgo sistémico, Cooperativismo & Desarrollo*. <https://doi.org/10.16925/co.v26i113.2194> [Consulta: 25 de mayo de 2023]

23) Taylor, J. (1998): "Cross-Industry Differences in Business Failure Rates: Implications for Portfolio Management." *Commercial Lending Review, p.* 36–46.

24) Yiu, T. (2021). *Understanding random forest*. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Consulta: 5 de abril de 2023]

ANEXO I. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**		X		
ODS 2. **Hambre cero.**		X		
ODS 3. **Salud y bienestar.**			X	
ODS 4. **Educación de calidad.**			X	
ODS 5. **Igualdad de género.**			X	
ODS 6. **Agua limpia y saneamiento.**			X	
ODS 7. **Energía asequible y no contaminante.**			X	
ODS 8. **Trabajo decente y crecimiento económico.**	X			
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades**.	X			
ODS 11. **Ciudades y comunidades sostenibles**.		X		
ODS 12. **Producción y consumo responsables.**		X		
ODS 13. **Acción por el clima.**			X	
ODS 14. **Vida submarina.**			X	
ODS 15. **Vida de ecosistemas terrestres.**			X	
ODS 16. **Paz, justicia e instituciones sólidas.**		X		
ODS 17. **Alianzas para lograr objetivos.**	X			

Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto:

1. ODS 8: Trabajo decente y crecimiento económico:

El modelo de riesgo de crédito ayuda a promover un crecimiento económico sostenible al evaluar de manera más precisa el riesgo crediticio de los clientes. Esto puede conducir a una asignación de recursos más eficiente y a una mayor estabilidad financiera.

2. ODS 9: Industria, innovación e infraestructura:

El modelo propuesto fomenta la innovación en el sector financiero al utilizar técnicas avanzadas de análisis de datos y aprendizaje automático para evaluar el riesgo crediticio. Además, puede contribuir a una infraestructura financiera más sólida al mejorar la gestión de riesgos en las instituciones bancarias.

3. ODS 10: Reducción de las desigualdades:

El modelo de riesgo de crédito más preciso y objetivo puede ayudar a reducir las desigualdades al proporcionar una evaluación justa y equitativa del riesgo crediticio para todos los solicitantes, sin sesgos ni discriminación.

4. ODS 17: Alianzas para lograr los objetivos:

La implementación y el uso efectivo del modelo de riesgo de crédito requerirán colaboración y alianzas entre instituciones financieras, reguladores y expertos en el campo. Esto promoverá una mayor cooperación para lograr los objetivos de desarrollo sostenible relacionados con la estabilidad financiera y el acceso justo al crédito.

ANEXO II. Información de variables

Tabla 4. Información extra de variables

Nombre de la variable	Descripción	Tipo	Media	Desv. Típica
RevolvingUtilizationOfUnsecuredLines	Balance de líneas, créditos, hipotecas… dividido entre la suma total de créditos	Porcentaje	5.89	257.04
age	Edad de la persona	Entero	51.28	14.42
DebtRatio	Pagos de deuda y vida divididos entre los ingresos totales	Porcentaje	26.6%	42%
MonthlyIncome	Ingresos mensuales	Real	6670,22	14384,67
NumberOfOpenCreditLinesAndLoans	Número de créditos y líneas abiertas como hipotecas, prestamos de coche…	Entero	8.75	5,17
NumberOfTimes90DaysLate	Veces que se ha retrasado 90 días o más en su pago	Entero	0.21	3,46
NumberRealEstateLoansOrLines	Número de hipotecas y préstamos incluyendo líneas de crédito del hogar	Entero	1,05	1,15
NumberOfDependents	Número de personas que dependen económicamente de la persona que solicita el crédito	Entero	0,85	1,14

Fuente: Elaboración propia.

ANEXO III. Código del modelo

CÓDIGO PARA ENTRENAR EL MODELO

```
import pandas as pd #para leer archivos CSV
import numpy as np #para arrays y operaciones matemáticas
import seaborn as sns #para ver datos y regresiones
import matplotlib.pyplot as plt #para sacar funciones por pantalla
from sklearn.metrics import accuracy_score #para la regresión logística
import pickle #para guardar el modelo


from sklearn.linear_model import LogisticRegression #para hacer la regresión logística
from sklearn.preprocessing import StandardScaler #para estandarizar datos
from sklearn.model_selection import train_test_split #para poder dividir los datos de
entrenamiento y test

#A continuación se leen los datos y se eliminan las filas que tengan "null"
datos_train = pd.read_csv("cs-training.csv",sep = ',',index_col=[0])

#train_limpios = datos_train.dropna()
datos_test = pd.read_csv("cs-test.csv",sep = ',',index_col=[0])
#test_limpios = datos_test.dropna()


#Saco un porcentaje de los datos que son de entreno y los que son de test
total_data = len(datos_train) + len(datos_test)
test_percent = len(datos_test)/total_data
train_percent = len(datos_train)/total_data
#imprimo
print("Porcentaje de datos de test: %.3f " % test_percent)
print("Porcentaje de datos de train: %.3f" % train_percent)



# Matriz de correlaciones.
```

```
# La matriz de correlación nos permite averiguar cómo de correladas o
# o relacionadas están dos variables. De este modo, podremos prescindir
# de campos cuya información sea explicada por otros campos.

corr = datos_train.corr()

# Inicializamos la firgura de matplotlib.
f, ax = plt.subplots(figsize=(7, 7))

cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Dibujamos el mapa de calor.
sns.heatmap(corr, cmap=cmap, vmax=1, center=0, vmin=-1,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})



#Dado que numerosas variables tienen una matriz de correlación igual a 1,
#eliminaremos una de las dos, ya que podemos incidir en problemas de colinealidad
#Las relaciones de 1 son:
# NumberOfTimes90DaysLate-NumberOfTIme30-59DaysPastDueNotWorse (elimino esta)
# NumberOfTimes90DaysLate-NumberOfTIme60-89DaysPastDueNotWorse (elimino esta)
#NumberOfTIme30-59DaysPastDueNotWorse-NumberOfTIme60-89DaysPastDueNotWorse
(ambas)

datos_train2=datos_train.drop(["NumberOfTime30-
59DaysPastDueNotWorse","NumberOfTime60-89DaysPastDueNotWorse"],axis=1)
datos_test2=datos_test.drop(["NumberOfTime30-
59DaysPastDueNotWorse","NumberOfTime60-89DaysPastDueNotWorse"],axis=1)



datos_train2=datos_train2.dropna() #elimina las filas con Nan




#pasamos ahora a entrenar el modelo y comprobarlo mediante regresión logística
```

```
X_train_data = datos_train2.drop("SeriousDlqin2yrs",axis=1) #esta linea permite quitar la columna primera y quedarse con el resto
y_train_data= datos_train2["SeriousDlqin2yrs"] #la columna quitada es la que se adjudica a esa variable
X_test = datos_test2.drop("SeriousDlqin2yrs",axis=1) #lo mismo para el test
X_test=X_test.dropna() #elimino las filas que tienen valores NaN
y_test = datos_test2["SeriousDlqin2yrs"] #la primera columna del test se adjudica a esa variable

X_train, X_val, y_train, y_val = train_test_split(X_train_data, y_train_data, random_state = 42)
#esto permite dividir los datos de la columna de forma aleatoria
#Las siguientes lineas sirven para estandarizar los datos
scaler = StandardScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)




# A continuación se realiza la regresión logística
lr = LogisticRegression(random_state=42, class_weight="balanced",max_iter=500) #se itera máximo 500 veces para que tenga fin
lr.fit(X_train_scaled, y_train)
y_pred = lr.predict(X_val_scaled)
accuracy = accuracy_score(y_val, y_pred)
print('Precisión: {:.2f}%'.format(accuracy*100))



#guardamos el modelo y el scaler
filename_model = 'finalized_model.sav'
filename_scaler = 'scaler.sav'
pickle.dump(lr, open(filename_model, 'wb'))
pickle.dump(scaler, open(filename_scaler, 'wb'))
```

```
r2_adj = 1 - (1 - accuracy) * (len(y_val) - 1) / (len(y_val) -
X_val_scaled.shape[1] - 1)
print('R^2 ajustado: {:.2f}%'.format(r2_adj*100))

print("\nEstadísticas descriptivas:")
print(datos_train2.describe().transpose())



X_train, X_val, y_train, y_val = train_test_split(X_train_data,
y_train_data, random_state=42)
scaler = StandardScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

X_train_sm = sm.add_constant(X_train_scaled)
logit_model = sm.Logit(y_train, X_train_sm)
logit_result = logit_model.fit()

# Calcular el valor absoluto del estadístico t para cada variable
t_values = np.abs(logit_result.tvalues[1:])  # Excluye la columna
constante

for variable, t in zip(X_train_data.columns,
np.abs(logit_result.tvalues[1:])):
    print(f"Variable: {variable}, |t|: {t}")



# A continuación se realiza la regresión logística
lr = LogisticRegression(random_state=42, class_weight="balanced",
max_iter=500)
lr.fit(X_train_scaled, y_train)
y_pred = lr.predict(X_val_scaled)

# Agrega una columna constante al conjunto de entrenamiento
X_train_scaled = sm.add_constant(X_train_scaled)

# Ajusta el modelo de regresión logística utilizando statsmodels
logit_model = sm.Logit(y_train, X_train_scaled)
result = logit_model.fit()
```

```
# Obtiene los p-values de los coeficientes del modelo
p_values = result.pvalues

# Imprime los p-values
print("P-Values del modelo:")
print(p_values)
```

CÓDIGO PARA GENERAR EL DASHBOARD

```
import dash
import pandas as pd
import pickle
from dash import dcc, html, Input, Output
from sklearn.preprocessing import StandardScaler
from dash.dependencies import Input, Output, State

app = dash.Dash(__name__)

# Cargamos el modelo y el scaler entrenados previamente con nuestros
datos:
filename_model = 'finalized_model.sav'
filename_scaler = 'scaler.sav'
model = pickle.load(open(filename_model, 'rb'))
scaler = pickle.load(open(filename_scaler, 'rb'))

app.layout = html.Div(
    style={'max-width': '600px', 'margin': 'auto', 'background-color':
'#f0f8ff', 'padding': '30px'},
    children=[
        html.H1('Riesgo de crédito TFG Pablo', style={'textAlign':
'center', 'marginBottom': '30px'}),
        html.H4('Ingresa los datos requeridos:'),
        html.Div(
            className='input-container',
            children=[
                html.Label('Ratio de utilización de líneas no
aseguradas'),
                dcc.Input(id='utilization-ratio', type='number',
placeholder='0-1', className='input-field')
            ]
        ),

        html.Div(
            className='input-container',
            children=[
                html.Label('Edad'),
                dcc.Input(id='edad', type='number', placeholder='20-80',
className='input-field')
            ]
```

```
        ),
        html.Div(
            className='input-container',
            children=[
                html.Label('Ratio de Deuda'),
                dcc.Input(id='debt-ratio', type='number', placeholder='0-
1', className='input-field')
            ]
        ),
        html.Div(
            className='input-container',
            children=[
                html.Label('Ingreso mensual'),
                dcc.Input(id='monthly-income', type='number',
placeholder='300-30.000', className='input-field')
            ]
        ),

        html.Div(
            className='input-container',
            children=[
                html.Label('Número de líneas de crédito y préstamos
abiertos'),
                dcc.Input(id='open-credit-lines', type='number',
placeholder='0-30', className='input-field')
            ]
        ),

        html.Div(
            className='input-container',
            children=[
                html.Label('Número de veces con retraso de pago de 90
días'),
                dcc.Input(id='payment-delay', type='number',
placeholder='0/1', className='input-field')
            ]
        ),
        html.Div(
            className='input-container',
            children=[
```

```
                html.Label('Número de préstamos o líneas de bienes
raíces'),
                dcc.Input(id='real-estate-loans', type='number',
placeholder='0-15', className='input-field')
            ]
        ),
        html.Div(
            className='input-container',
            children=[
                html.Label('Número de dependientes'),
                dcc.Input(id='dependents', type='number', placeholder='0-
5', className='input-field')
            ]
        ),
        html.Button('Ejecutar', id='submit-button', n_clicks=0,
className='button'),
        html.Div(id='result-container', style={'marginTop': '30px'})
    ]
)

@app.callback(
    Output('result-container', 'children'),
    [Input('submit-button', 'n_clicks')],
    [State('utilization-ratio', 'value'),State('edad', 'value'),
State('debt-ratio', 'value'), State('monthly-income', 'value'),
     State('open-credit-lines', 'value'),
     State('payment-delay', 'value'), State('real-estate-loans',
'value'),
     State('dependents', 'value')]
)
def predict(n_clicks,  utilization_ratio, age,
debt_ratio,  monthly_income,  open_credit_lines,
             payment_delay, real_estate_loans, dependents):
    if n_clicks > 0:
        print("Valores de las variables:")
        print("utilization_ratio:", utilization_ratio)
        print("age:", age)
        print("debt_ratio:", debt_ratio)
        print("monthly_income:", monthly_income)

        print("open_credit_lines:", open_credit_lines)
```

```
        print("payment_delay:", payment_delay)
        print("real_estate_loans:", real_estate_loans)
        print("dependents:", dependents)

        try:
            age = float(age)
            monthly_income = float(monthly_income)
            debt_ratio = float(debt_ratio)
            open_credit_lines = float(open_credit_lines)

            data = {

                'RevolvingUtilizationOfUnsecuredLines':
[utilization_ratio],
                'age': [age],
                'DebtRatio': [debt_ratio],
                'MonthlyIncome': [monthly_income],

                'NumberOfOpenCreditLinesAndLoans': [open_credit_lines],
                'NumberOfTimes90DaysLate': [payment_delay],
                'NumberRealEstateLoansOrLines': [real_estate_loans],
                'NumberOfDependents': [dependents]
            }

            df = pd.DataFrame(data=data)
            df_scaled = scaler.transform(df)  # Aplicar el escalado al
nuevo dato ingresado
            predicted = model.predict(df_scaled)

            if predicted[0]:
                result_text = '¡Felicidades! Se te ha concedido el
crédito.'
                result_color = 'green'
            else:
                result_text = 'Lo sentimos, no cumples los requisitos
para obtener el crédito.'
                result_color = 'red'

            result_message = html.Div(
                style={'textAlign': 'center', 'marginTop': '20px'},
                children=[
```

```
                    html.H3(result_text, style={'color': result_color})
                ]
            )

            return result_message

        except Exception as e:
         error_message = html.Div(
            style={'textAlign': 'center', 'marginTop': '20px'},
            children=[
             html.H3('Ocurrió un error en el procesamiento de los datos:
{}'.format(str(e)), style={'color': 'red'})
             ]
        )
         return error_message

    else:
        return ''


if __name__ == '__main__':
    app.run_server(debug=True)
```