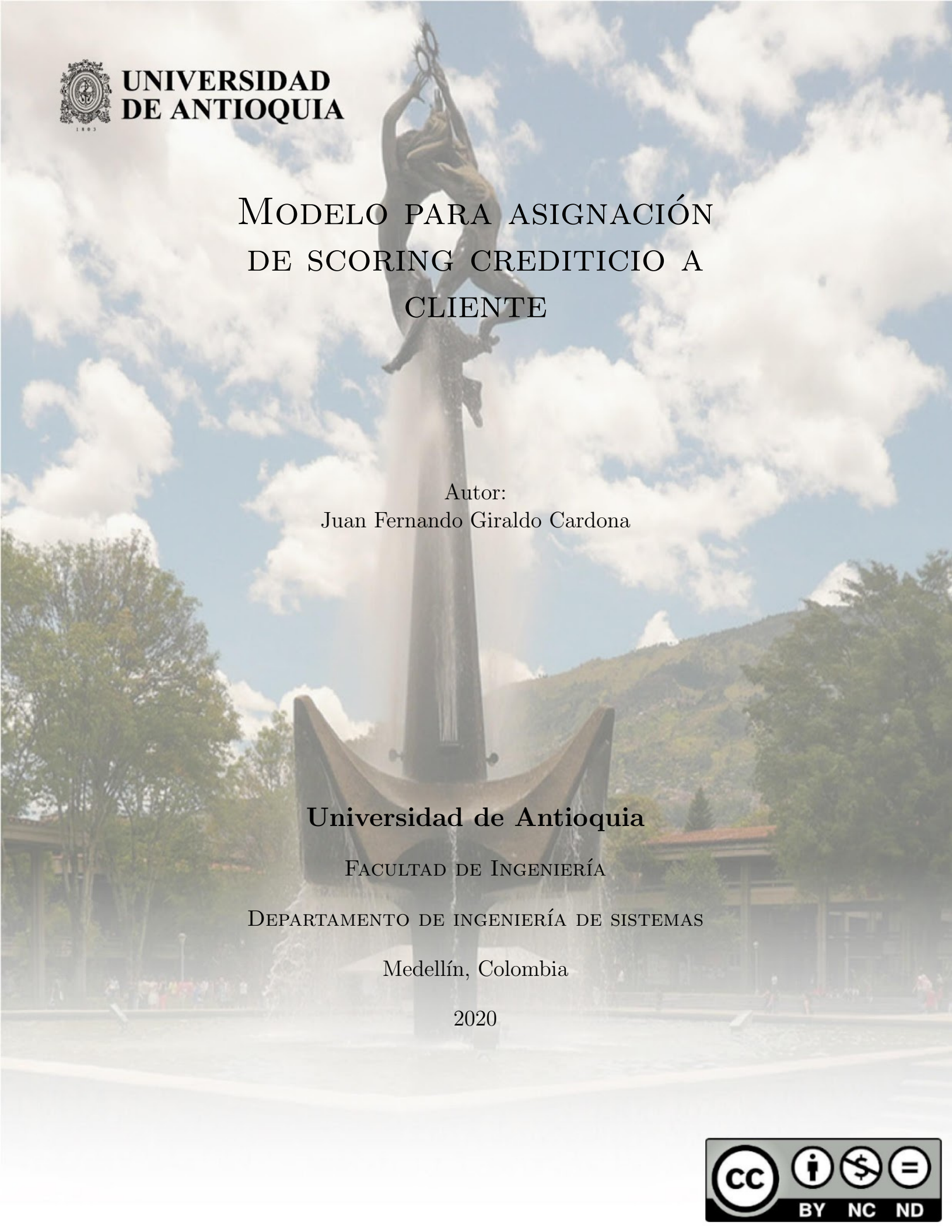

UNIVERSIDAD DE ANTIOQUIA

Modelo para asignación de scoring crediticio a cliente

Autor:
Juan Fernando Giraldo Cardona

Universidad de Antioquia

Facultad de Ingeniería

Departamento de ingeniería de sistemas

Medellín, Colombia

2020

Modelo para asignación de scoring crediticio a cliente

Informe práctica empresarial como requisito parcial para optar al título de: ingeniero de sistemas

Autor:
Juan Fernando Giraldo Cardona

Asesores
Javier Fernando Botia Valderrama
Sebastián Arango Munoz
Alejandro Castaño Rojas

Universidad de Antioquia

Facultad de Ingeniería

Departamento de ingeniería de sistemas

Medellín, Colombia

2020

Índice

Índice de figuras

Índice de tablas

Lista de acrónimos		
Acrónimo	**Significado**	**Significado en español**
ML	Machine learning	Aprendizaje automático
ROC	Receiver operating characteristic	Curva de característica operativa del recepto
AUC	Area under the ROC curve	Área bajo la curva ROC
SVM	Support vector machine	Máquinas de soporte vectorial
GBM	Gradient Boosting Machine	
AWS	Amazon web services	
SAM	Serverless Application Model	

1. Resumen

El crecimiento de las investigaciones y producción de software para la solución de problemas de aprendizaje automático ha hecho que empresas de todo tipo y tamaño se interesen y vean esto como una oportunidad para la mejora de sus procesos en algunas áreas. Un ejemplo claro es que la adopción de los modelos de asignación de scoring crediticio como herramienta de ayuda para la toma de decisiones en las áreas de evaluación y aceptación crediticia ha aumentado en los últimos años. En este proyecto se busca entrenar un modelo de aprendizaje automático utilizando los datos de ventas e historiales de pago que la empresa @PC MAYORISTA ha almacenado de sus clientes en su sistema de gestión empresarial. Inicialmente se hace una extracción, limpieza, etiquetado y preprocesamiento de los datos, luego se entrenan diferentes algoritmos de aprendizaje automático para comprar cuál es el que mejor se ajusta a los datos y mejores predicciones hace con el conjunto de datos de prueba, donde se evidencia que los algoritmos basados en árboles como Gradient Boosting Machine (GBM por sus siglas en inglés) y XGBoost son los que mejores predicciones hacen sobre los datos en los diferentes conjuntos de validación. Después se hace una propuesta de una arquitectura en la nube donde se pueda alojar este modelo ya entrenado y finalmente se implementa un prototipo en una versión beta que servirá a los empleados de la empresa para hacer pruebas y verificar su correcto funcionamiento y posibles mejoras.

Palabras clave: Modelo de scoring crediticio, aprendizaje automático, arquitectura serverless, clasificación, toma de decisiones.

2. Introducción

Últimamente se ha visto cómo las nuevas tecnologías aplicadas en diferentes áreas han facilitado los procesos de muchas empresas que se han atrevido a implementarlas. El aprendizaje de máquina (Machine learning o ML) es una rama de la computación que busca que los computadores actúen sin ser programados de manera explícita, la cuál se ha venido utilizando ampliamente. En la última década el aprendizaje de máquina ha hecho posible la creación de autos que se manejan solos, dispositivos de reconocimiento de audio, búsquedas más efectivas en la web y un mejor entendimiento del genoma humano. [2] El sector económico como la banca y otras entidades crediticias también se han visto en la necesidad de usar este tipo de tecnología ya que al recolectar datos de sus clientes como información personal, transacciones, etc. durante mucho tiempo, han tenido que desarrollar sistemas de análisis autónomos los cuáles sean de ayuda a la hora de hacer actividades como supervisión del mercado, detección de patrones en los balances financieros, evolución de los contratos hipotecarios, estudios de transacciones fraudulentas e incluso evaluación y asignación de puntaje crediticio. [5]

Según informes de la compañía *TransUnion*, *"en Colombia se ha visto un crecimiento en el crédito de consumo durante los últimos años debido a que las entidades financiera están otorgando créditos para todos los perfiles de riesgo pero de una manera controlada"*. Para las entidades es importante asegurar que el aumento en la aprobación de créditos no se convierta luego en un aumento en los casos de morosidad de los consumidores ya que esto afecta de manera directa la rentabilidad de estas entidades. El acceso a los créditos debe garantizarse en todos los sectores de la población pero de una manera moderada. Esta tarea se hace difícil por el cargo operativo y el tiempo que se toma ya que las empresas deben tener personal disponible que se encargue de la gestión e investigación en centrales de riesgo para obtener el perfil del usuario que aplica al crédito. Esta es una de las razones por las cuales compañías que operan en Colombia como Bancolombia, Davivienda, Movistar, Claro, Suramericana, han abierto sus puertas a la investigación y posteriormente adopción de tecnologías basadas en inteligencia artificial que le sirvan de ayuda en la optimización y realización eficiente de estas actividades.

El grupo empresarial @PC Mayorista tiene como objetivo brindar servicios de aprovisionamiento de hardware, licenciamiento, suministros y servicios de infraestructura. La empresa le brinda la posibilidad a sus clientes de acceder a los productos y servicios por medio de créditos que ellos deben solicitar. En la empresa @PC Mayorista hay un equipo encargado de evaluar las peticiones de crédito que hacen los clientes. Inicialmente deben recolectar información relevante que ayude a la toma de decisiones como ciudad donde están ubicados, ingresos por venta del año inmediatamente anterior, historial de compra que el cliente tiene con la empresa, información adicional que ha obtenido el asesor de venta a cargo como el cupo del crédito y el plazo de pago. Otro requisito es que toda la información que le den a la empresa debe ser verificable de alguna manera. Como proceso adicional, el equipo encargado busca las referencias cre-

diticias del cliente aspirante al crédito en plataformas como Datacrédito. Luego de este proceso una persona se encarga de consignar toda la información en una tabla de Excel donde todos pueden acceder a ella y posteriormente hacer una evaluación conjunta dónde basados en ciertas métricas y la experiencia, deciden si el cliente tiene la capacidad de endeudamiento respecto al cupo y plazo de pago solicitado para finalmente denegar o aprobar el crédito. Muchas veces este proceso se vuelve engorroso y difícil de llevar a cabo ya que requiere un estudio muy detallado del cliente y de su caso puntual, lo cual conlleva a que las personas encargadas del proceso deban dedicar más tiempo del debido en esta actividad y retrasen otras. Se evidencia un sobre esfuerzo de las capacidades laborales de cada empleado. Otro problema que se puede evidenciar es que la empresa ha venido aumentando su adquisición de clientes en los últimos periodos, esto abre la posibilidad de que las solicitudes para aperturas de crédito aumenten y con ello la empresa @PC MAYORISTA deba contratar más personal para aumentar la capacidad en su área de estudio y aprobación crediticia que pueda atender la alta demanda de solicitudes.

Con ayuda de herramientas de software se propone utilizar los datos de los clientes que la empresa @PC MAYORISTA ha recolectado para implementar un modelo de score crediticio el cual se integre a su lógica de negocio y ayude a las personas encargadas de determinar la aprobación crediticia de los clientes, a tomar una decisión teniendo en cuenta los resultados que este modelo arroje.

3. Estado del arte

En los últimos años se han vuelto muy populares las plataformas de préstamos en línea, sobre todo en países como Estados Unidos y China. Un prestatario puede enviar fácilmente una solicitud de préstamo incluyendo una gran cantidad de información que estas plataformas solicitan. Estas plataformas suelen usar varias herramientas de control de riesgo para rechazar prestatarios no calificados. Lending Club, una de las organizaciones más grandes dedicada al préstamo entre pares, sólo ha aceptado aproximadamente el 9 % de las solicitudes de préstamo que recibe en su plataforma. Li Zhiyong, Tian Ye, et al [1], han visto esto como una oportunidad para realizar un trabajo de investigación e implementación que tiene como objetivo principal usar un modelo basado en máquinas de soporte vectorial llamado "Máquinas de soporte vectorial semi-supervisadas"(SSVM) que hace uso de las muestras no etiquetadas para obtener información adicional sobre la estructura de los datos. Por lo tanto, se espera que funcione mejor que la SVM tradicional cuando el conjunto de datos contiene muestras no etiquetadas, especialmente para el caso que solo contiene una pequeña proporción de puntos etiquetados en el conjunto de entrenamiento. Con la ayuda de este modelo buscan resolver el problema de rechazo inferencial (reject inference) que se origina en las soluciones basadas en aprendizaje de máquina orientadas a obtención de puntaje crediticio para clientes que aplican a créditos. La mayoría de estos modelos se entrenan utilizando sólo la información de los clientes que les han aceptado sus solicitudes de crédito porque los prestamistas usualmente

no almacenan la información de aquellos clientes a los cuales les ha sido negado. Por esta razón estos modelos presentan un sesgo en sus muestras. Para medir el rendimiento de este algoritmo usan métricas como matriz de confusión, curva ROC y posteriormente lo comparan con otros modelos como regresión logística y SVM utilizando la métrica del el área bajo la curva ROC (AUC) donde finalmente se muestra en una tabla comparativa que el modelo SSVM es el que tiene mejores resultados y encuentran que las ventajas de SSVM sobre los otros dos modelos provienen principalmente de las solicitudes aceptados con etiquetas verdaderas, lo que significa que SSVM puede hacer uso de la información en los rechazados para mejorar la clasificación en los aceptados.

Otro grupo de investigadores conformado por Cuicui Luo, Desheng Wu, Dexiang Wu [10], de la escuela de negocios de la Universidad de Estocolmo en Estocolmo, Suecia., se interesó también en evaluar el rendimiento de los algoritmos de aprendizaje de máquina como lo son regresión logística, SVM y perceptrón multicapa (MLP por su nombre en inglés multilayer perceptron), que son aplicados comúnmente a soluciones de puntaje crediticio; con el objetivo de comparar sus resultados con otro modelo que ellos proponen basado en redes neuronales llamado redes de creencias profundas (DBN por su nombre en inglés "deep belief networks") el cuál, según mencionan en su artículo, nunca ha sido objeto de estudio para evaluar el riesgo crediticio. Uno de los objetivos principales de la investigación es proporcionar un conjunto de pruebas y resultados que sienten las bases para investigaciones y/o trabajos futuros en DBN aplicado al puntaje crediticio en mercados CDS (credit default swaps) o "permuta de cobertura por incumplimiento crediticio"por su traducción al español que es básicamente un seguro que cubre a su tenedor del riesgo de impago de un préstamo o de la compra de otro producto financiero. DBN es un modelo generativo multicapa el cual se obtiene mediante el entrenamiento de varias capas basadas en máquinas de Boltzmann restringidas con el cual buscan descubrir posible información oculta en el conjunto de datos. Para minimizar la dependencia e incrementar la fiabilidad del modelo se usó validación cruzada con k igual a 10 en el entrenamiento de cada uno de los experimentos con los modelos. Luego aplicaron métricas como AUC y eficiencia en la clasificación sobre los modelos, el mejor resultado lo obtuvo el modelo DBN con un 87.75 % de eficiencia en la clasificación seguido por SVM con 87.4 % y con la métrica AUC DBN sigue con los mejores resultados y esta vez MLP tiene un mejor resultados que SVM.

Por último se referencia una investigación muy completa titulada "Extreme Learning Machines for Credit Scoring: An Empirical Evaluation" [3] donde sus autores se interesan, como su título lo indica, en evaluar un modelo basado en redes neuronales llamado máquinas de aprendizaje extremo (ELM) para resolver problemas de asignación de puntaje crediticio. Las ELM son modelos basados en las redes neuronales con propagación hacia adelante y sus nodos de la capa oculta son elegidos de manera aleatoria. En particular el aprendizaje de un ELM es equivalente a estimar un modelo de regresión lineal o logística desde el punto de vista computacional, ya que en teoría, este algoritmo tiende a proporcionar un buen rendimiento de generalización a una velocidad de aprendizaje extremadamente rápida. ELM sólo tiene dos hiperparámetros que son el

número de neuronas en la capa oculta y su función de activación. En la investigación plantean una evaluación más detallada de los modelos ya que además de hacer comparaciones entre el desempeño de estos (allí calculan métricas como eficiencia y AUC), también miden características como facilidad de uso, donde se tiene en cuenta el número de hiperparámetros y la sensibilidad del modelo respecto a estos, y complejidad computacional que se resume particularmente a consumo de tiempo computacional y uso de memoria. Para las pruebas usaron algoritmos que han sido utilizados ampliamente para la solución de este tipo de problemas, como lo son regresión logística regularizada (LR-R), SVM (con kernel lineal y radial), k vecinos más cercanos (KNN), árboles de decisión (J48 Y CART); también se propuso redes neuronales artificiales como mayor competidor ya que estos son conocidos en las aplicaciones financieras por sus buenos resultados, y por último el modelo que ellos proponen, máquinas de aprendizaje extremo. Para la evaluación se emplearon tres conjuntos de datos diferentes, los cuales contienen diferentes grupos de variables de tipo financiero, demografía social, etc. En términos de facilidad de uso, se puede observar que el modelo LR-R tiene los mejores resultados y se destaca que, a pesar de que ELM quedó dentro de los modelos con menos facilidad de uso, este presenta una estabilidad mayor que ANN. Queda en evidencia que estos algoritmos requieren una mayor optimización en los hiperparámetros que otros como SVM o KNN. La medición del tiempo computacional se llevó a cabo sacando los tiempos promedio de entrenamiento de cada modelo en milisegundos (ms) y el uso de memoria en kb dentro de cada uno de los 3 conjuntos de datos. El que menos memoria usa es KNN pero debido a que la diferencia entre este y los otros clasificadores es mínima, se concluye que todos los algoritmos en términos del uso de memoria. Para el caso de tiempo de computación el algoritmo predictivo más eficiente para cada conjunto de datos es LR-R, aunque vale a pena mencionar que la discrepancia en el consumo de tiempo entre LR-R, ELM, KNN, CART y J4.8 es mínima, ANN y SVM-L por el contrario requieren sustancialmente más tiempo. En la evaluación de rendimiento no se detecto un algoritmo que se presentara un resultado que marcara una gran diferencia entre los demás, sin embargo cabe resaltar que SVM-R fue un poco mejor en la mayoría de los casos de prueba. Finalmente cabe resaltar que ELM es tan competitivo como ANN ya que en las pruebas se evidencian resultados muy parecidos o incluso mejores. Estas investigaciones aportan muchas ideas de los modelos que se usan para resolver problemas de puntaje crediticio acompañado de las técnicas que algunos entusiastas emplean para mejoras un poco los resultados en términos de rendimiento y complejidad computacional, acompañado de las métricas de validación empleadas. Se evidencia también que para estos problemas no es necesario el uso de técnicas de aprendizaje de máquina muy avanzadas, con algoritmos demasiado robustos para obtener buenos resultados a la hora de aplicarlos a un proyecto empresarial.

4. Objetivos

4.1. Objetivo general

Plantear una solución basada en machine learning la cual permita analizar la información de los clientes de la empresa @PC MAYORISTA que se ha recolectado previamente en la base de datos, con el fin de saber si estos son elegibles o no para un crédito y que sirva como herramienta de apoyo a los empleados que llevan a cabo la tarea de estudiar las solicitudes de crédito que hacen los clientes dentro de la empresa.

4.2. Objetivos específicos

- Analizar la base de datos de la empresa para encontrar características y relaciones importantes que brinden información relevante al modelo.
- Caracterizar la base datos para la correcta implementación del modelo.
- Aplicar un modelo de machine learning el cual presente resultados con una eficiencia mínima del 70 %, que sirvan como una medida de confianza para toma de decisiones, y a su vez usar métricas para validar la efectividad del modelo.
- Proponer una adecuada estrategia de visualización con la que los usuarios de negocio puedan hacer uso del modelo desarrollado.

5. Marco teórico

El aprendizaje de máquina o machine learning (en inglés) se define como un conjunto de métodos que pueden detectar patrones en datos de manera automática y luego usarlos para tomar decisiones o predecir datos futuros.[15] Existen diferentes tipos de aprendizaje de máquina:

5.1. Aprendizaje supervisado

Trabaja con un conjunto de datos etiquetados. Por cada muestra en el conjunto de entrenamiento se tiene un objeto de entrada y uno de salida. En el aprendizaje supervisado las variables de salida pueden ser categóricas o discretas, cuando son categóricas se conoce como un problema de clasificación y cuando son discretas se conoce como un problema de regresión.

5.2. Aprendizaje no supervisado

Deja que el algoritmo encuentre un patrón oculto en una carga de datos. Con el aprendizaje no supervisado no hay respuestas correctas o incorrectas, es cuestión de ejecutar el algoritmo y ver qué patrones y resultados ocurren [9]

5.3. Aprendizaje por refuerzo

Se busca que el software tome ciertas decisiones basado en recompensas o castigos que se le dan durante su entrenamiento.

5.4. Clasificación

Es tal vez la forma de aprendizaje de máquina más ampliamente usada. El objetivo de la clasificación es asignarle una clase y a una muestra de entrada x. La clasificación binaria ocurre cuando el número de clases o etiquetas es igual a dos, si existen más clases se denomina como clasificación multiclase. [15]

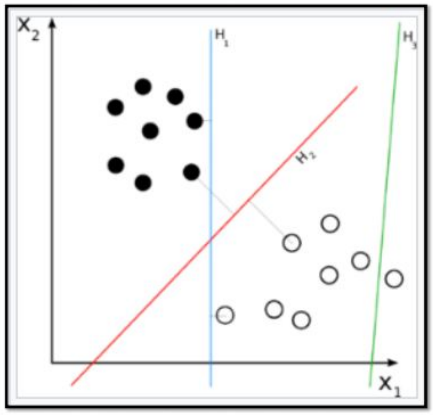

Figura 1: Ejemplo de clasificación binaria.
[16]

La clasificación tiene numerosas aplicaciones, va desde detección de fraude, campañas de mercado hasta el diagnósticos médicos y psicológicos, etc. Durante un proceso de clasificación se llevan a cabo dos fases principales. La primera fase o fase de entrenamiento es cuando el modelo de clasificación construye al clasificador utilizando el conjunto de datos de entrenamiento para esto, el cuál es un subconjunto del total de muestras y clases (o etiquetas) que se están analizando. Esta fase también se puede entender como el mapeo de una función $y = f(X)$ que describa la forma de los datos y que pueda predecir, con cierta eficiencia, la clase y asociada a una muestra X. En la segunda fase el modelo se usa para la clasificación, en primer lugar utiliza la función hallada en la primer fase para predecir la clase a la que pertenecen las muestras que se le ingresen, en este caso se usa el conjunto de prueba que también es un subconjunto del total de muestras y clases diferente al conjunto de entrenamiento. Luego de obtener la clasificación de cada muestra se calcula la eficiencia predictiva del clasificador, este valor se obtiene al comparar el valor predicho con el valor real, en caso de que los resultados sean aceptables se puede seguir utilizando este modelo para futuras muestras. [7]

5.5. Clustering

Se clasifica como un método de aprendizaje no supervisado y se define básicamente como organizar un grupo de objetos que comparte características similares. Se usa ampliamente en áreas como el marketing ya que puede agrupar clientes en segmentos, las redes sociales lo usan para determinar comunidades de usuarios, etc.[9] K-Means, DBSCAN y HDBSCAN son algunos de los algoritmos de aprendizaje no supervisado que se suelen usar para resolver problemas donde las muestras no tienen ninguna clase (o etiqueta) asignada, al no tener una, cambia la forma en que se hace la validación de las predicciones en este tipo de modelos. En las validaciones internas se buscan dos características principalmente:

- **Compactación:** mide qué tan cerca están las muestras dentro del mismo clúster una poca variación interna de los datos es un buen indicio de buena compactación,
- **Separabilidad:** mide qué tan separados están los clústers entre sí. Para obtener este tipo de métrica se puede calcular la distancia entre los centros de cada clúster y adicionalmente la distancia que hay entre las muestras de cada uno.

Generalmente las métricas que se usan para la validación miden compactación y separabilidad, por ejemplo está la métrica de la silueta que mide principalmente qué tan bien asignado está cada muestra. $S(i)$ (donde i es una muestra del conjunto de datos) varia entre -1 y 1, si el valor es -1 la muestra no pertenece al clúster que fue asignada, si el valor es 0 significa que los clústers se sobreponen y si el valor es 1 es porque la muestra está bien asignada. [4]

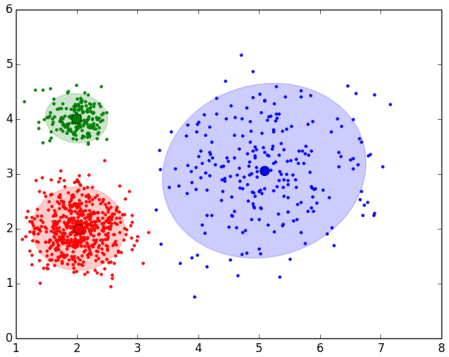

Figura 2: Representación gráfica de un clúster.
[14]

5.6. Modelo

Un modelo se crea aplicando un algoritmo a los datos, pero es más que un algoritmo o un contenedor de metadatos, es un conjunto de datos, estadística y patrones que pueden ser aplicados a nuevos datos para generar predicciones y hacer inferencias sobre relaciones.[12]

5.7. Limpieza de datos

Los datos del mundo real tienden a ser incompletos, ruidosos e inconsistentes. Las rutinas de limpieza de datos intentan completar los valores faltantes, suavizar el ruido al identificar valores atípicos y corregir las inconsistencias en los datos.[7]

5.8. Detección de datos atípicos

Es un procedimiento cuyo propósito es identificar los parámetros que se en afectados por datos atípicos existentes dentro del conjunto de datos. Un datos atípico es una muestra que se desvía notablemente de las otras muestras pertenecientes al mismo conjunto de datos.[8] Técnicamente, elegir la medida de similitud/distancia y el modelo de relación para describir los objetos de datos es fundamental en la detección de valores atípicos. Desafortunadamente, estos a menudo dependen de la aplicación. Las diferentes aplicaciones pueden tener requisitos muy diferentes. La alta dependencia de la detección de valores atípicos del tipo de aplicación hace que sea imposible desarrollar un método de detección de valores atípicos aplicable universalmente. En su lugar, se deben desarrollar métodos de detección de valores atípicos individuales dedicados a aplicaciones específicas. [7]

5.9. Ingeniería de características

Es el acto de extraer características de datos sin procesar y transformarlas en formatos que sean adecuados para el modelo de aprendizaje de máquina. Es un paso crucial en el proceso de aprendizaje de máquina, porque las características correctas pueden aliviar la dificultad de modelar y, por lo tanto, permitir que el proceso genere resultados de mayor calidad.[17]

5.10. Estandarización de variables

Algunos datos tienen límites en sus valores pero hay otros que pueden incrementar su valor sin límite, este comportamiento puede afectar ciertos modelos. Como su nombre lo sugiere, la estandarización de variables cambia la escala de los datos a un rango en particular, en algunos casos ayuda a optimizar los cálculos en un algoritmo. Hay diferentes tipos de operaciones de estandarización:

5.10.1. Escalado estándar

Este método consiste en restar la media de la variable a cada dato de entrada x y se divide por su desviación estándar (raíz cuadrada de la varianza). La variable resultante tiene una media igual a 0 y una varianza igual a 1.

$$\bar{x} = \frac{x - mean(x)}{\sqrt{var(x)}} \tag{1}$$

5.10.2. Escala min-max

Este método obtiene los valores mínimos y máximos de la variable dentro de todo el conjunto de datos y luego comprime o estira, según el caso, todos los valores de la variable para que estén dentro de un rango de [0, 1]. Uno de los problemas que tiene este método es que sí la variable tiene ruido, se amplía.

$$\bar{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

5.11. Reducción de dimensionalidad

Dimensionalidad en aprendizaje de máquina se conoce como el número de variables de entrada o características en un conjunto de datos. La reducción de dimensionalidad se refiere a deshacerse de los datos que brindan poca información al modelo y conservar la que es crucial. Hay muchos métodos para llevar a cabo reducciones de dimensionalidad y cada uno define sus propias métricas para saber qué tanta información aporta una variable al modelo.[17] Cuando se tratan conjuntos de datos con alta dimensionalidad, es muy útil reducir su dimensionalidad proyectando los datos a un subespacio dimensional menor que capture la esencia de los datos. [15] Tener un espacio de características muy grande significa también que probablemente el volumen de datos es alto y no es estrictamente cierto que mientras más datos se ingresen a un modelo mas información va a obtener. Tener muchos datos puede impactar el rendimiento del modelo y su capacidad de generalización por lo tanto es preferible tratar de conservar las variables que más información aportan.

5.12. t-SNE (t-distributed Stochastic Neighbor Embedding)

Es una técnica de reducción de dimensionalidad que se usa principalmente con el objetivo de visualizar la distribución de los datos haciendo una proyección de estos en subespacios de una dimensión menor dándole a cada muestra una ubicación en un subespacio de 2 o 3 dimensiones. Esta técnica es una variación de SNE (por sus siglas en inglés Stochastic Neighbor Embedding) [6] que es mucho más fácil de optimizar, y produce visualizaciones significativamente mejores reduciendo la tendencia de juntar las muestras en el centro del subespacio. t-SNE es mejor que las técnicas existentes al momento de crear un subespacio que proyecte la distribución de los datos en diferentes dimensiones. [11]

5.13. Selección de características

Las técnicas de selección de características buscan eliminar características inútiles con el fin de reducir la complejidad del modelo. El objetivo final es un modelo simple que sea rápido de calcular con poca o ninguna degradación en la eficiencia predictiva.[17] Para alcanzar un modelo tal, algunas técnicas de selección de características necesitan entrenar más de un modelo candidato. En otras palabras, la selección de características no se trata de reducir el tiempo de entrenamiento; de hecho, algunas técnicas aumentan el tiempo total de entrenamiento, sino de reducir el tiempo que tarda el modelo en arrojar un resultado. En términos generales, las técnicas de selección de características se dividen en tres clases:

5.13.1. Métodos de filtrado

Las técnicas de filtrado preprocesan características para eliminar aquellas que probablemente no sean útiles para el modelo. Estas técnicas son mucho menos costosas que las técnicas de envoltorio pero no tienen en cuenta el modelo empleado. Por lo tanto, es posible que no puedan seleccionar las características correctas para el modelo. Es mejor hacer un prefiltrado conservador, para no eliminar involuntariamente funciones útiles antes de que lleguen al paso de entrenamiento del modelo.

5.13.2. Métodos de envoltorio (wrapper)

Estas técnicas son costosas, pero permiten probar subconjuntos de características, lo que significa que no eliminará accidentalmente características que no son informativas por sí mismas pero que son útiles cuando se usan en combinación. Está técnica trata al modelo como una caja negra que proporciona una puntuación de calidad de un subconjunto propuesto para las características.

5.13.3. Métodos integradas (embedded)

Estos métodos realizan la selección de características como parte del proceso de capacitación del modelo. Los métodos integrados incorporan la selección de características como parte del proceso de capacitación del modelo. No son tan poderosos como los métodos de envoltura, pero no son tan costosos. En comparación con el filtrado, los métodos integrados seleccionan características que son específicas del modelo. En este sentido, los métodos integrados logran un equilibrio entre el gasto computacional y la calidad de los resultados.

5.14. Métricas para la validación del modelo

Es el proceso en el cual se evalúa un modelo entrenado en el conjunto de prueba. [17] Para evaluar el modelo se pueden emplear diferentes métricas que miden cuán acertado es el modelo prediciendo las muestras del conjunto de entrenamiento versus las nuevas muestras del conjunto de prueba o validación.

5.14.1. Matriz de confusión

Es una herramienta muy útil para analizar qué tan bien puede el modelo clasificar las muestras en sus respectivas clases. Las columnas de una matriz de confusión representan lo que el algoritmo predice y las columnas corresponden a los valores reales, para entender mejor la información que se presenta en las filas y columnas de la matriz de confusión, es necesario definir algunas medidas:

- **Verdaderos positivos** *(TP)* **:** se refiere a las muestras positivas que fueron clasificadas correctamente.
- **Verdaderos negativos** *(TN)* **:** se refiere a las muestras negativas que fueron clasificadas correctamente.
- **Falsos positivos** *(FP)* **:** son las muestras que se etiquetaron incorrectamente como positivas.
- **Falsos negativos** *(FN)* **:** el conjunto de muestras positivas que se etiquetaron incorrectamente como negativas.

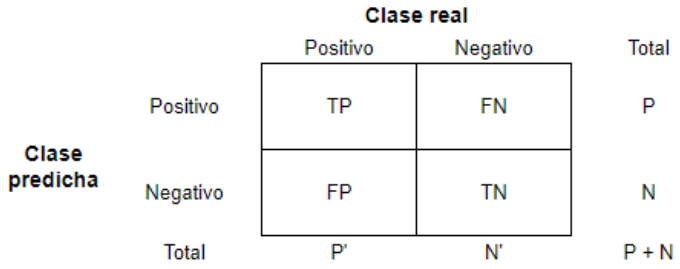

Figura 3: Matriz de confusión con resultados totales de las muestras positivas y negativas.

La matriz de confusión de la Figura 3 ejemplifica un problema de clasificación binaria donde P' es el número de muestras que fuero etiquetadas como positivas $TP + FP$ y N' es el número de muestras que fueron etiquetadas como negativas $TN + FN$. El número total de muestras se puede obtener de $TP + TN + FP + FN$ o $P + N$ o $P' + N'$. Cabe aclarar que una matriz de confusión se puede aplicar fácilmente a un problema de múltiples clases. [7]

5.14.2. Eficiencia (accuracy)

Es el porcentaje del total de muestras clasificadas correctamente, representa qué tan bien el algoritmo asigna las muestras a las diferentes clases dentro del problema.

$$eficiencia = \frac{TP + TN}{P + N} \tag{3}$$

5.14.3. Exactitud (precision)

Es el porcentaje de muestras predichas que pertenecen a cierta clase que en realidad sí pertenecen a esa clase.

$$exactitud = \frac{TP}{TP + FP} \tag{4}$$

5.14.4. Sensibilidad (recall)

Se le conoce también como *tasa de verdaderos positivos* y representa el porcentaje de muestras positivas que fueron identificadas correctamente por el clasificador.

$$sensibilidad = \frac{TP}{TP + FN} \tag{5}$$

5.14.5. Especificidad

Se le conoce también como *tasa de verdaderos negativos* y representa el porcentaje de muestras negativas que fueron identificadas correctamente por el clasificador.

$$especificidad = \frac{TN}{TN + FP} \tag{6}$$

5.14.6. Medida F (F measure)

Es una alternativa para usar la exactitud y la sensibilidad de manera combinada, también conocida como medida F_β, donde β es un número real positivo. Da igual peso a la exactitud y a la sensibilidad, mientras mas alto sea el valor de la medida F, mejor es la capacidad de predicción del modelo en cuestión.

$$F_\beta = \frac{(1 + \beta^2) * exactitud * sensibilidad}{exactitud + sensibilidad} \tag{7}$$

5.14.7. Validación cruzada k-fold

Se define como una técnica de remuestreo iterativa, que consiste esencialmente en hacer una división del conjunto de datos en k subconjuntos, donde se eligen k-1 subconjuntos para hacer el entrenamiento del modelo y se deja uno para hacer la validación. Este proceso se repite k veces, donde se cumple que cada subconjunto es utilizado una vez para hacer la validación del modelo, por lo general se recomienda usar un valor de k entre 5 y 10. Este método presenta ciertas ventajas, la principal es que maneja un mejor balance entres el sesgo (bias) y la varianza, por otro lado el costo computacional es considerablemente bajo. Este método es muy usado para hacer selección de modelos comparando el error en cada uno. El error se calcula como:

$$CV_k = \frac{1}{k} \sum_{i=1}^{k} (Err_i) \tag{8}$$

5.15. Aprendizaje automático (AutoML)

Es el proceso de automatizar el tiempo de consumo de las tareas iterativas en el desarrollo de un modelo de aprendizaje de máquina. Esto permite a los científicos de datos, analistas, desarrolladores, etc. Construir modelos más eficientes y productivos manteniendo siempre la calidad de este. Lo que lo diferencia con el desarrollo de modelos basados en aprendizaje de máquina tradicional es que consumen menos recursos, no requieren un gran conocimiento y tiempo para producir y comprar varios modelos. Uno de sus principales objetivos es acelerar el tiempo que toma tener un modelo muy eficiente listo para producción.

En la etapa de entrenamiento los modelos basados en aprendizaje automático crea una cantidad de tuberías que funcionan de forma paralela y en cada una se prueban diferentes algoritmos y parámetros. El servicio itera a través de algoritmos emparejados con selecciones de características, donde cada iteración produce un modelo con una puntuación de entrenamiento. Cuanto mayor sea la puntuación, mejor se considerará que el modelo "se ajusta.ª sus datos. Se detendrá una vez que alcance los criterios de salida definidos en el experimento o cuando alcance un tiempo máximo de ejecución, esto depende de cómo se especifique en el experimento.

El aprendizaje automático no sólo sirve para seleccionar el algoritmo que mejores métricas obtenga, también se puede aplicar en la etapa de ingeniería de características (featurization), allí se aplican un conjunto de técnicas para normalizar y estandarizar los datos, se hace cargo de los datos faltantes y finalmente hace la codificación de las variables categóricas.[13]

6. Metodología

Con la intención de llevar a cabo un proceso ordenado durante la realización del proyecto se decidió aplicar una metodología la cual segmentara el proyecto en fases secuenciales, y se busca cumplir un objetivo principal en cada una. El proyecto se va a dividir en 3 fases principales que se explicarán a continuación: La primera es **recolección y preprocesamiento de los datos**, que consiste en pedir los accesos para conexión remota a la base de datos de réplica, que está alojada en un data center local de la empresa @PC MAYORISTA. Luego se debe analizar los datos que se tienen, su dimensión y las diferentes relaciones que hay en la base de datos, para posteriormente hacer un proceso de limpieza de los datos ya que es muy importante saber si hay datos faltantes, eliminar los que estén duplicados o que sean irrelevantes y no aporten mucha información al sistema.

En la segunda fase, se va a realizar la **búsqueda del modelo** que sea mejor para nuestros datos. Inicialmente se debe evaluar si de acuerdo a los datos resultantes de la fase anterior se puede aplicar un modelo de aprendizaje supervisado o aprendizaje no supervisado. En caso de que sea supervisado se debe verificar si, de acuerdo a la solución que se busca, el problema es de clasificación o de regresión. En caso de que sea no supervisado se buscarán métodos de

agrupamiento (clustering).

La tercera fase consiste en **entrenar y evaluar el modelo con los datos.** Inicialmente se debe dividir el conjunto de datos en 3 subconjuntos, los cuales son, conjunto de entrenamiento (training set), el cual, como su nombre lo indica se usa para entrenar el modelo elegido en la fase anterior; conjunto de validación (validation/dev set) se usa para ajustar los parámetros del modelo; y conjunto de prueba (test set) con el cual se evalúa el desempeño del modelo y sólo es usado al final del proceso, luego de que el modelo haya finalizado el ciclo de entrenamiento. También se usarán técnicas como matriz de confusión para evaluar qué tan bien entrenado está el modelo y su eficiencia de acuerdo a los criterios definidos por el área de negocio.

Finalmente se busca proponer una estrategia de visualización para que los empleados de la empresa @PC MAYORISTA puedan hacer uso del modelo. Para esto se debe desplegar el modelo en un servidor, ya sea en el local de la empresa o utilizado algún servicio de internet en la nube. Esta decisión se va a discutir con el equipo de negocio y el equipo de T.I.

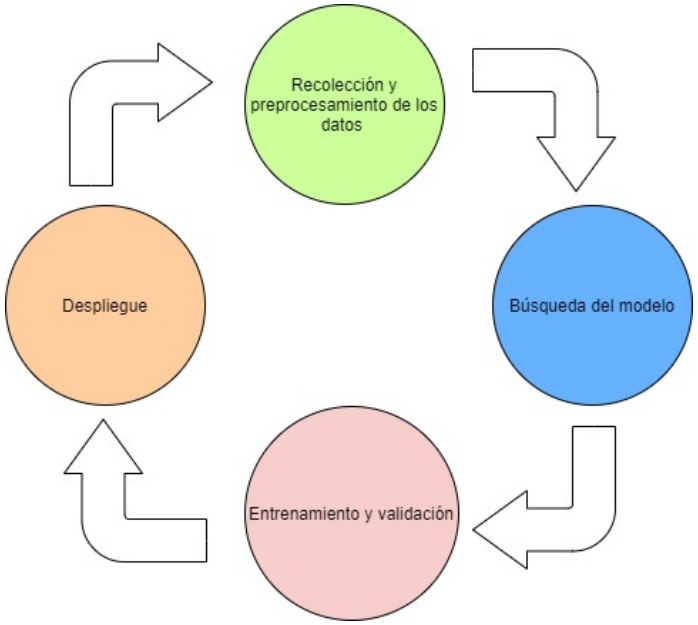

Figura 4: Ciclo de vida del proyecto

A continuación se hará una breve descripción de los procesos realizados para cumplir los objetivos planteados en cada una de las fases descritas con anterioridad:

6.0.1. Recolección y exploración de los datos

Este es el proceso de mayor importancia en el proyecto porque de este depende en gran medida los resultados que se obtendrán con los modelos. Inicialmente se hace la solicitud a la empresa @PC MAYORISTA y a sus administradores de base de datos para acceder a la información de los clientes, ellos hacen una réplica de la base de datos y nos brindan los accesos necesarios para hacer las consultas requeridas sobre esta, todo bajo un acuerdo de confidencialidad puesto que allí hay cierta información sensible almacenada. En ese momento se inicia una exploración de toda la base de datos relacional que está gestionada por la tecnología Microsoft SQL server. Una percepción inicial es que la base de datos contiene diferentes problemas, por ejemplo no hay documentación acerca de su diseño, implementación y uso, por ello es necesario hacer un proceso de ingeniería inversa para obtener el diagrama de entidad relación y lograr un panorama más general sobre las tablas, sus relaciones, e información almacenada. Posteriormente se encuentra que hay mucha información con un sistema de nombramientos un poco difícil de comprender y con registros redundantes, por ello se solicita un acompañamiento temporal por parte de uno de los encargados del área de evaluación y aceptación crediticia ya que sería de gran ayuda por su conocimiento y habilidad para obtener reportes de estos datos. Después de varias sesiones con esta persona se adquiere una gran compresión sobre la estructura de la base de datos y como resultado una consulta (query) que obtiene la información más relevante de los clientes. Los resultados de esta consulta se almacenan en un archivo de extensión csv y se importa en un notebook con kernel configurado en Python 3.7 para iniciar con la limpieza y transformación de los datos.

6.0.2. Limpieza y transformación de los datos

En principio se obtiene un conjunto de datos con 350888 muestras y 16 variables, lo cuál es una cantidad considerable de información a procesar, por ello se procede con la verificación de muestras para saber si hay algunas que están repetidas y posteriormente eliminarlas, después de este proceso se observa que hay una reducción del conjunto del 33.19 %, lo cual indica que una cantidad considerable de los datos estaban duplicados. También se hace una depuración sobre las columnas que tienen información no relevante para el sistema como ids. Luego se examina el tipo de dato con el que han sido reconocidas las variables y se hacen las debidas correcciones sobre cada uno. Finalmente se hace una verificación de datos faltantes que arroja resultados muy satisfactorios porque todas las muestras tienen la información de cada una de las variables. También se cambia el nombre de las columnas para obtener un conjunto de variables con nombres más descriptivos. Luego de todo este proceso queda como resultado un conjunto de datos con 234420 muestras y 17 variables. Un dato muy importante a tener en cuenta es que el conjunto de datos no estaba etiquetado pero gracias a la información dada por la persona de @PC MAYORISTA se pudo detectar cuáles clientes habían estado en mora con el pago de sus créditos, así que se procede a

hacer el etiquetado de las muestras y se obtiene un conjunto de datos con dos clases, una representa los clientes que han estado en mora y otra representa a los clientes que no han estado en mora, como se muestra en la Figura 5.

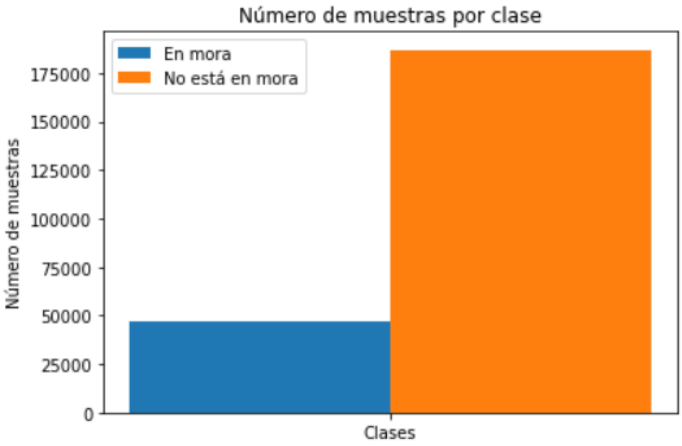

Figura 5: Numero de muestras por clase

Al observar la figura se puede notar que se tienen clases bastante desbalanceadas, la clase con clientes en mora contiene 47210 muestras mientras que la clase con clientes que no han estado en mora contiene 187210 muestras. Es probable que esto sea un problema a la hora de entrenar y probar el modelo sobre el conjunto de datos ya que no habrá un proceso de generalización, en otras palabras, el algoritmo va a aprender más sobre la clase más representativa y esto conlleva a una mala clasificación de la información en la fase de prueba. Antes de utilizar el conjunto de datos resultante para entrenar el algoritmo es necesario llevar a cabo un proceso de codificación de características (variables), en otras palabras, convertir las variables categóricas en numéricas debido a que esto hace que nuestro algoritmo entienda las variables de este tipo y pueda procesar la información, también aumenta la eficiencia de este, finalmente se lleva a cabo un proceso de estandarización de las variables. Durante este proceso también se lleva a cabo un proceso de detección de outliers empleando el algoritmo de clustering HDBSCAN pero se descarta porque se encuentra que para este problema elimina información importante para el modelo y causa aumento en el error de entrenamiento. Con la ayuda del algoritmo t-SNE se hace una reducción de la dimensión del conjunto de datos para hacer una visualización de la distribución de los datos, en la Figura 6 se puede evidenciar esta distribución, la cual evidentemente presenta cierta pérdida de información a causa de la reducción, y de nuevo el problema de desbalance de clases que se va a tratar más adelante.

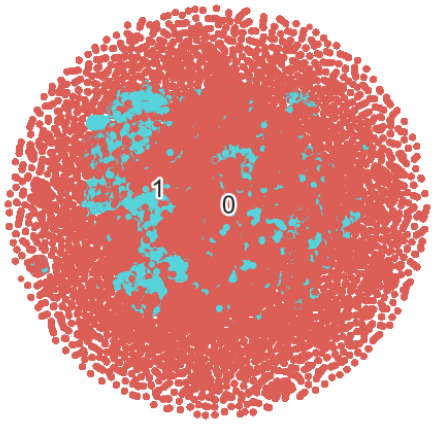

Figura 6: Distribución del conjunto de datos en 2D

6.0.3. Búsqueda del modelo

Usando la información encontrada en el estado del arte se encontró que es común el uso de algoritmos clásicos para el tratamiento de estos problemas, por eso se toma la decisión de empezar a probar con algoritmos como regresión logística, máquinas de soporte vectorial (kernel lineal) y random forest, también se busca hacer la comparación entre la eficiencia de los algoritmos cuando se aplica la técnica de selección de características y cuando no. Para preparar los datos se aplica un escalamiento min-max y luego se divide en tres subconjuntos distribuidos de la siguiente manera 80 % para entrenamiento, 10 % para validación y 10 % para probar al final de todo el proceso de elección e implementación. Después se emplea el algoritmo de creación de muestras artificiales SMOTE sobre la clase minoritaria pero sólo en el subconjunto de entrenamiento. Para el entrenamiento de los algoritmos se hace uso de la librería Scikit-learn 0.23.1, y para la elección de características se hace uso de la librería Mlxtend 0.17.2 y se emplea la técnica de selección secuencial hacia adelante. Las métricas usadas fueron matriz de confusión y medida F con $\beta = 1$ tanto en el conjunto de entrenamiento como en el da validación.

Para encontrar el mejor modelo también se hizo uso de una librería llamada h20 3.30.1.1 la cuál fue de gran ayuda para saber cual era el mejor algoritmo que resuelve el problema con los datos suministrados. Para esta implementación se hizo el acondicionamiento de una instancia virtual en la nube de Google (GCP) porque este método es bastante exigente a nivel computacional. Después de obtener el mejor algoritmo se procede a hacer una búsqueda de los parámetros más óptimos por medio de la técnica Grid Search, finalmente, al tener un modelo que al ser evaluado con las métricas de validación cumplía con los resultados esperado, se exporta en formato .pkl para su uso posterior.

Modelo	Sin selección de características		Con selección de características	
	Eficiencia	Medida F	Eficiencia	Medida F
Regresión logística	99.96 %	0.99	99.57 %	0.99
SVM	100 %	1	100 %	1
Random forest	100 %	1	100 %	1

Tabla 1: Comparación entre métricas de modelos con y sin selección de características

6.0.4. Estrategia de uso y visualización

Para que el personal del área de evaluación y aceptación crediticia pueda hacer uso del modelo entrenado se propone una arquitectura serverless sencilla en la cual se pueda montar un prototipo en una función lambda que se active cada vez que el usuario, en este caso una persona del área de estudio y aprobación crediticia, haga una petición HTTP por medio de una interfaz gráfica de una aplicación web. Aprovechando el partnership que la empresa tiene con AWS, se hace el montaje de esta infraestructura en el servicio de nube que este proveedor ofrece.

7. Resultados y análisis

Al hacer las evaluaciones sobre cada uno de los modelos mencionados, se obtienen resultados sorprendentemente buenos en todas las pruebas realizadas, tanto en el conjunto de entrenamiento como en el de pruebas. En la Tabla 1 se pueden observar los resultados de las métricas sobre los conjuntos de validación de cada uno de los modelos

A decir verdad es un poco extraño que todos los modelos tengan tan buen rendimiento. Inicialmente se descarta que sea un caso de sobreajuste ya que tanto en entrenamiento como en validación los resultados son similares, entonces se propone hacer una matriz de correlación para analizar qué tan fuerte es la relación de cada variable de entrada con la variable de salida ya que posiblemente algunas variables están sesgando al modelo.

Después de analizar un poco la Figura 7 se puede notar que hay una alta correlación entre la variable de entrada cupo y la variable de salida (clase) mora, también es notable la correlación entre la variable de entrada plazo y la variable de salida mora. Esto indica que al momento de etiquetar el conjunto de datos utilizando estas dos variables se incurrió en un descuido al no intuir que estas variables iban a sesgar al modelo. Es por ello que se vuelve a hacer toda la fase de entrenamiento y validación de los algoritmos pero esta vez sin hacer uso de las variables cupo y plazo que presentan una alta correlación debido al proceso de etiquetado. Esta vez, como se puede observar en la Tabla 2 los resultados fueron totalmente opuestos en lo que a eficiencia y la métrica F1 respecta.

Después de un disminución considerable en el rendimiento de los modelos se

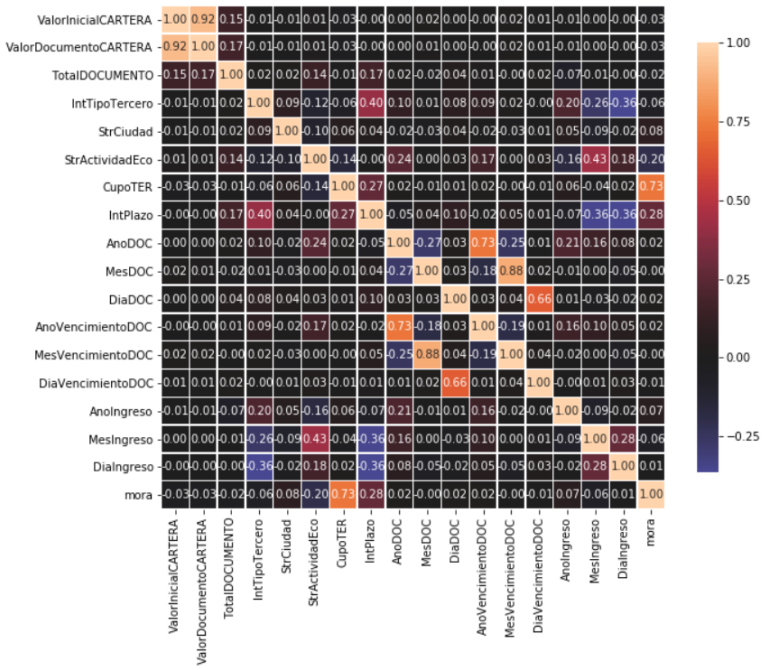

Figura 7: Matriz de correlación

toma la decisión de emplear una funcionalidad que la herramienta H2O proporciona para analizar la importancia de las variables de entrada que están siendo utilizadas para entrenar el modelo. Al encontrar que algunas variables tenían una importancia relativa muy baja se eliminan del conjunto de datos y se vuelve a ejecutar la etapa de entrenamiento. Esta vez, con ayuda de la librería XGBoost v1.3.0, se usa un algoritmo basado en árboles llamado XGBoost el cual es una mejora del algoritmo Gradient Boosting Machine (GBM por sus siglas en inglés) donde se busca explotar los recursos computacionales para obtener unos mejores resultados. Inicialmente se tuvo una eficiencia del 96.78 % y un F1 de 0.81 en el conjunto de validación, lo cual demuestra una gran capacidad de predicción del modelo, sin embargo, con la ayuda de Grid Search se hace una estimación de parámetros para tratar de optimizar el modelo y mejorar un poco más los resultados, luego de correr varias veces el algoritmo para ajustar parámetros se obtiene finalmente un modelo con una eficiencia de 99.98 % en el conjunto de validación y un F1 de 0.99. Se elige este modelo como candidato para usarlo en la etapa de implementación.

Como ya se mencionó la arquitectura para la aplicación donde el modelo está alojado, tiene un enfoque serverless. Como se muestra en la figura 8 el modelo se aloja en una capa de la lambda y se llama cada vez que se hace una petición al servicio desde el aplicativo web. Una de las dificultades que se presenta es que las lambdas sólo tienen una capacidad de 260 MB y al momento

Modelo	Sin selección de características		Con selección de características	
	Eficiencia	Medida F	Eficiencia	Medida F
Regresión logística	79.01 %	0.01	86.57 %	0.34
SVM	9.72 %	0.17	62.35 %	0.32
Random forest	9.72 %	0.17	67.76 %	0.20

Tabla 2: Comparación entre métricas de modelos con y sin selección de características

de hacer el montaje del backend con las dependencias se nota que la librería XGBoost es bastante pesada. Inicialmente se piensa en cambiar el diseño de la arquitectura a una orientada a contenedores pero teniendo en cuenta que se buscaba probar el modelo en un prototipo poco costoso se toma la decisión de volver a entrenar el modelo usando el algoritmo GBM, el cual se define como un algoritmo de ensamble, esto significa que va a obtener un modelo final basado en un conjunto de modelos individuales donde se combinan una serie de modelos denominado débiles con el objetivo de obtener un resultado mejor. Para este algoritmo también se hizo un proceso de optimización de hiperparámetros para lograr unos buenos resultados. Una ventaja de usar este algoritmo es que está implementado en la librería Sklearn la cuál tiene un peso tal que puede utilizarse sin problema en una función lambda.

Finalmente se continúa con el desarrollo del frontend utilizando la tecnología Angular v10.0, donde el usuario pueda agregar sin problema la información para obtener el score crediticio de 1 o hasta 100 clientes según las pruebas de rendimiento realizadas sobre el servidor en AWS.

Cabe aclarar que todo el montaje de la infraestructura se llevó a cabo utilizando la tecnología de infraestructura como código (IaC) de AWS llamada Serverless Application Model(SAM por sus siglas en inglés) el cual es un framework open source que provee sintaxis simples en YAML para la construcción de aplicaciones. Este método facilita la construcción y migración de la infraestructura.

8. Conclusiones

- Los resultados obtenidos en este proyecto son bastante satisfactorios ya que cumplen con los objetivos que se propusieron inicialmente, además se observa lo útil que es para una empresa como @PC MAYORISTA que tiene un flujo constante de clientes, disponer de una herramienta que le ayude a sus empleados en el proceso de evaluación y aprobación crediticia.

- Para este caso de uso los modelos basados en árboles han mostrado ser bastante buenos a la hora de presentar una solución para este tipo de problemas.

- A la hora de utilizar una métrica para definir la eficiencia de un modelo de

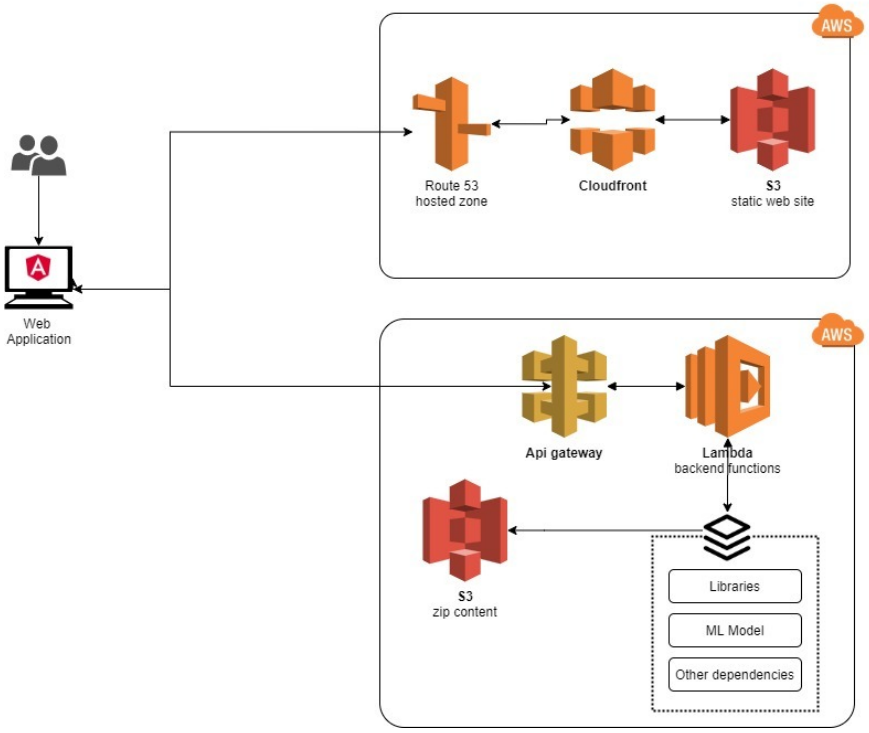

Figura 8: Arquitectura serverless propuesta

clasificación, es recomendable tener en cuenta las clases y la importancia que tiene cada una para la solución del problema.

- Usar tecnologías como el autoML ayudan bastante a la obtención a de resultados que de manera rápida ayudan a obtener una idea de los modelos que probablemente tengan mejor desempeño para tratar de resolver el tipo de problema que se está tratando.
- El estado del arte es un insumo fundamental para hacer un sondeo de las técnicas y recursos que otros investigadores ya han empleado, De esta forma se facilita el planteamiento de una hoja de ruta a seguir durante la ejecución del proyecto y se evita perderse en la inmensidad de caminos que se pueden seguir para solucionar el problema que se plantea.
- Emplear herramientas de infraestructura como código IaC es una muy buena opción a tener en cuenta a la hora de levantar una infraestructura desde cero o de crear nuevos módulos para esta ya que puede ser fácilmente modificada o migrada o otro proveedor de nube en caso de que se requiera (teniendo en cuenta la complejidad de la infraestructura).

- La ejecución de proyectos de aprendizaje automático se compone de diferentes fases que a su vez forman un proceso cíclico en el que el equipo encargado tendrá que repetir en varias ocasiones el mismo proceso con el objetivo de refinar los resultados del modelo. Lo más importante es entender el problema y principalmente saber el propósito de la solución para evitar perder el foco durante el proceso.

9. Recomendaciones

1. Como ya se mencionó anteriormente la fase de ingeniería de características es una de las más importantes en un proyecto de ML ya que es allí donde se prepara al conjunto de datos para su modelado. Por lo aprendido en la ejecución de este proyecto se sugiere experimentar con varias técnicas y algoritmos teniendo como objetivo principal facilitar el entendimiento de la estructura de los datos y con ello lograr que los modelos descubran patrones en los datos e información valiosa.

2. Encontrar el modelo que mejor describa el comportamiento de los datos es también una fase de gran importancia durante el proceso de ejecución del proyecto. El estado del arte es un recurso muy útil en esta etapa porque ilustra cómo otros investigadores han abordado este tipo de problemas, qué tipos de modelos han aplicado en sus soluciones y qué problemas han tenido y cómo los han abordado, con esta información ya se acota un poco la variedad de algoritmos que se pueden usar para el entrenamiento del modelo. Otro aspecto a tener en cuenta es que esta fase suele ser muy exigente a nivel de recursos computacionales, si se usan recursos propios es posible que se tome bastante tiempo en obtener un resultado de todos los algoritmos que se prueban, por ello se recomienda usar herramientas alojadas en la nube como Google Colaboratory o AWS Sagemaker que disponen de bueno recursos, optimizados para este tipo de tareas y ayudan un poco en la ejecución de esta fase.

3. Para arquitecturas serverless montadas en recursos como Lambda de AWS donde el tamaño es muy limitado se deben utilizar estrategias como verificar las librerías que se están utilizando qué funciones se requieren de cada una. En caso de que se exceda el límite del tamaño permitido se pueden utilizar técnicas como borrar algunos archivos o disminuir la versión de algunas librerías cuidando que las funciones empleadas no cambien su definición

Referencias

[1] Li Zhiyong et al. «Reject inference in credit scoring using Semi-supervised Support Vector Machines, Expert Systems With Applications». En: *Expert Systems with Applications* (2017). DOI: 10.1016/j.eswa.2017.01.011.

[2] Ng Andrew. *Machine Learning*. 2017. URL: https://www.coursera.org/learn/machine-learning.

[3] Lessmann Stefan Beque Artem. «Extreme Learning Machines for Credit Scoring: An Empirical Evaluation». En: *Expert Systems with Applications* (2017). DOI: 10.1016/j.eswa.2017.05.050.

[4] et al Brock Guy. *clValid: An R Package for Cluster Validation*. 2008. URL: https://www.jstatsoft.org/article/view/v025i04.

[5] Andreas Joseph Chiranjit Chakraborty. *Machine learning at central banks*. 2017. URL: https://www.bankofengland.co.uk/working-paper/2017/machine-learning-at-central-banks.

[6] Hinton Geoffrey y Roweis Sam. *Stochastic Neighbor Embedding*. 2002. URL: https://www.cs.toronto.edu/~fritz/absps/sne.pdf.

[7] Pei Jian Han Jiawei Kamber Micheline. «Data Mining Concepts and Techniques». En: Elsevier Inc, 2017. Cap. 3. ISBN: 978-0-12-381479-1.

[8] Ching-Hsien Hs Hui-Huang Hsu Chuan-Yu Chang. «Big Data Analytics for Sensor-Network Collected Intelligence». En: Academic Press, 2017. Cap. 13. ISBN: 978-0-12-809393-1. DOI: http://dx.doi.org/10.1016/B978-0-12-809393-1.00001-5.

[9] Bell Janson. «Machine Learning: Hands-On for Developers and Technical Professionals». En: John Wiley y Sons, Inc., 2015. Cap. 1-8. ISBN: 978-1-118-88906-0.

[10] Wu Dexiang Luo Cuicui Wu Desheng. «A deep learning approach for credit scoring using credit default swaps». En: *Engineering Applications of Artificial Intelligence* (2016). DOI: https://dx.doi.org/10.1016/j.engappai.2016.12.002.

[11] van der Maaten Laurens e Hinton Geoffrey. *Visualizing Data using t-SNE*. 2008. URL: https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdff.

[12] Microsoft. *Mining Models*. 2018. URL: https://docs.microsoft.com/en-us/analysis-services/data-mining/mining-models-analysis-services-data-mining.

[13] Microsoft. *What is automated machine learning (AutoML)?* 2020. URL: https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml.

[14] Ricardo Moya. *Selección del número óptimo de Clusters*. 2016. URL: https://jarroba.com/wp-content/uploads/2016/06/Cluster3C.png.

[15] Murpy Kevin P. «Machine Learning A Probabilistic Perspective». En: Massachusetts Institute of Technology, 2012. Cap. 1.1. ISBN: 978-0-262-01802-9.

[16] SimpliLearn. *Classification - Machine Learning*. 2018. URL: https://www.simplilearn.com/classification-machine-learning-tutorial#:~:text=Classification%3A%20Meaning,an%20input%20variable%20as%20well..

[17] Casari Amanda Zheng Alice. «Feature Engineering for Machine Learning». En: O'Reilly Media, Inc., 2018. Cap. 2. ISBN: 978-1-491-95324-2.