UNIVERSIDAD NACIONAL AGRARIA LA MOLINA

FACULTAD DE ECONOMÍA Y PLANIFICACIÓN

"ESTIMACIÓN DEL INGRESO DE NO CLIENTES DE UNA ENTIDAD FINANCIERA MEDIANTE ALGORITMOS XGBOOST Y CATBOOST"

PROYECTO DE TRABAJO DE SUFICIENCIA PROFESIONAL PARA OPTAR TÍTULO DE INGENIERA ESTADÍSTICA INFORMÁTICA

MARILUNA OLARTE AREVALO

LIMA, PERÚ

2025

ÍNDICE

I. Introducción

1.1. Problemática

El ingreso de una persona constituye un componente esencial en los procesos de evaluación crediticia que realiza la entidad financiera, pues refleja de manera objetiva el nivel de recursos económicos percibidos por el solicitante y permite identificar su estabilidad financiera. Contar con una estimación precisa del ingreso facilita conocer con claridad la situación económica de cada persona y dimensionar adecuadamente su nivel de ingresos. Según la normativa vigente, "el otorgamiento del crédito está determinado por la capacidad de pago del solicitante, que a su vez está definida fundamentalmente por su flujo de ingresos netos de gastos y obligaciones crediticias y sus antecedentes crediticios" (Superintendencia de Banca, Seguros y AFP, 2023, p. 7), lo que resalta la importancia de disponer de información confiable sobre los ingresos al evaluar solicitudes de crédito.

En este contexto, la entidad requiere contar con un estimador de ingresos que le permita aproximar de manera confiable el nivel de ingresos de personas que no son clientes, debido a que no existe una relación previa con estos solicitantes. Disponer de este estimador es fundamental para sustentar las decisiones de otorgamiento de préstamos y reducir la incertidumbre en el proceso de evaluación crediticia. No obstante, la ausencia de vínculo previo con estas personas implica que la entidad no cuente con variables internas, como registros de transacciones, historial de pagos o patrones de uso de productos, que permitan validar de forma directa sus ingresos. Por esta razón, la entidad se ve limitada a emplear únicamente variables externas, tales como datos demográficos y de comportamiento crediticio provenientes del Reporte Consolidado de Créditos, las cuales, si bien aportan información relevante, pueden no capturar con exactitud la realidad económica de cada solicitante (Villaseca et al., 2020).

Una estimación por debajo del nivel real de ingresos puede excluir a solicitantes con capacidad económica suficiente, limitando su acceso al crédito, mientras que una sobreestimación incrementa el riesgo de incumplimiento y puede deteriorar la calidad de la cartera de la entidad financiera (James et al., 2021). Disponer de una estimación precisa no solo es fundamental para una adecuada gestión del riesgo crediticio, sino también para diseñar estrategias comerciales más efectivas, ya que permite segmentar correctamente a los potenciales clientes y ofrecerle productos ajustados a su realidad económica, en términos de montos, tasas y plazos (Gutiérrez et al., 2019). Además, la normativa vigente exige que las entidades cuenten con mecanismos adecuados para determinar los ingresos antes de otorgar un crédito, siendo este un aspecto supervisado por la Superintendencia de Banca, Seguros y AFP como parte de una gestión prudente del riesgo (SBS, 2023). En este contexto, se plantea la necesidad de desarrollar un modelo de estimación de ingresos para personas que no son clientes, utilizando algoritmos de boosting como XGBoost y CatBoost, que permitan aproximar de manera confiable el nivel de ingresos a partir de variables externas disponibles.

1.2. Marco teórico

La Superintendencia de Banca, Seguros y AFP (SBS) del Perú establece que la adecuada evaluación de los ingresos de los solicitantes de crédito es un elemento central en la gestión de riesgos de las entidades financieras. Según la SBS, conocer y estimar de forma razonable los ingresos permite a las entidades determinar la capacidad de pago real del cliente, asegurando que las obligaciones crediticias asumidas puedan ser atendidas con su flujo de ingresos habitual. Esto contribuye a la protección del sistema financiero, evitando la concentración de riesgos y el sobreendeudamiento de los clientes. Además, la SBS indica que la estimación de ingresos es fundamental para clasificar correctamente a los deudores y calcular de forma adecuada las provisiones, tal como se establece en el Reglamento para la Evaluación y Clasificación del Deudor y la Exigencia de Provisiones (Resolución SBS N° 11356-2008 y sus modificatorias). De esta manera, la entidad puede mantener un portafolio saludable, cumplir con las disposiciones regulatorias y garantizar la sostenibilidad de las operaciones crediticias.

Con el avance de la banca de consumo, surgió la necesidad de estimar los ingresos de clientes con condiciones laborales menos estructuradas, como aquellos con ingresos informales, variables o sin documentación tradicional. Esta realidad llevó a las entidades financieras a incorporar modelos paramétricos, como regresiones lineales, que permitieran aproximar sus ingresos utilizando variables indirectas como el nivel de ventas del negocio, el consumo de servicios básicos o el número de dependientes. Sin embargo, en la última década, el proceso de digitalización del sistema financiero y el crecimiento en la disponibilidad de datos alternativos han dado paso a técnicas más avanzadas, que han demostrado ser más eficaces para capturar patrones complejos de comportamiento económico. Estas metodologías no solo han incrementado la precisión de las estimaciones de ingreso, sino que también han contribuido a una evaluación crediticia más dinámica, permitiendo incorporar a segmentos históricamente excluidos del sistema financiero formal (Berg et al., 2020; World Bank, 2024).

Una de las técnicas que se ha incorporado en los últimos años es el Machine Learning o aprendizaje automático, una rama de la inteligencia artificial enfocada en el desarrollo de algoritmos y modelos capaces de identificar patrones a partir de datos, sin requerir instrucciones programadas de manera explícita para ejecutar una tarea específica (Mitchell, 1997). Mediante el uso de esta técnica, los sistemas pueden procesar grandes volúmenes de información, reconocer relaciones entre variables y emplear ese conocimiento para generar predicciones o apoyar la toma de decisiones de manera automatizada. En el ámbito financiero, estas capacidades permiten estimar ingresos de clientes, evaluar riesgos crediticios y optimizar procesos de originación de créditos, utilizando de manera eficiente los datos disponibles. El machine learning permite que una máquina “aprenda de la experiencia”, mejorando su desempeño a medida que recibe nuevos datos, y este aprendizaje puede clasificarse en tres enfoques principales: supervisado, no supervisado y por refuerzo, según el tipo de datos empleados y la forma en que el modelo aprende.

El aprendizaje supervisado se aplica cuando se cuenta con una variable objetivo conocida, permitiendo que el modelo aprenda la relación entre las variables independientes y dicha variable para predecir resultados en nuevos datos, como en la estimación de ingresos de clientes o la predicción de la probabilidad de incumplimiento de pago en el sector financiero. Por su parte, el aprendizaje no supervisado se utiliza cuando no se dispone de una variable objetivo,

enfocándose en descubrir patrones, estructuras ocultas o relaciones dentro de los datos, siendo útil para segmentar clientes según comportamientos similares o detectar anomalías en transacciones. Finalmente, el aprendizaje por refuerzo se basa en la interacción con un entorno mediante ensayo y error, donde el modelo recibe recompensas o penalizaciones en función de sus acciones, aprendiendo a maximizar una recompensa acumulada, con aplicaciones en estrategias de trading algorítmico y en la optimización de procesos en banca.

En el contexto del aprendizaje supervisado, una de las técnicas más utilizadas es el gradient boosting, la cual permite construir modelos de alta precisión para predicción de variables objetivo a partir de otras variables de entrada. Esta técnica consiste en la construcción secuencial de predictores, donde cada nuevo modelo se entrena para corregir los errores del modelo anterior (Friedman, 2001). En cada iteración, se calcula el gradiente (la dirección del error) de la función de pérdida y se ajusta un nuevo árbol de decisión para reducir ese error, logrando que el conjunto de modelos trabaje de manera conjunta para mejorar de forma progresiva la precisión de las predicciones.

1.2.1 Modelo XGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de aprendizaje supervisado que se basa en la técnica de gradient boosting, la cual construye modelos de manera secuencial para mejorar las predicciones de un modelo anterior. En cada paso, XGBoost ajusta un árbol de decisión para corregir los errores cometidos por el modelo previo, minimizando el error general del conjunto. Una de las características más destacadas de XGBoost es su optimización de segundo orden, que utiliza información de la derivada segunda de la función de pérdida, lo que permite mejorar la precisión y la rapidez del proceso de entrenamiento en comparación con otros métodos. Además, es capaz de manejar valores faltantes de manera eficiente, lo que le permite tomar decisiones sobre cómo manejar estos valores sin necesidad de imputarlos previamente. Con estas características, es conocido por su alto rendimiento, eficiencia computacional y su capacidad para abordar problemas complejos, siendo ampliamente utilizado en tareas de clasificación y regresión (Chen & Guestrin, 2016).

Modelo de Predicción

El modelo de XGBoost se construye como una combinación secuencial de árboles de decisión. Para una entrada x la predicción $\hat{y}(x)$ de un modelo XGboost con T árboles se expresa como:

$$\hat{y}(x) = \sum_{t=1}^{T} f_t\,(x)$$

Donde:

- $\boldsymbol{f_t(x)}$ es el $\boldsymbol{t}$ -ésimo árbol del modelo.

- La función $\boldsymbol{f_t(x)}$ es generalmente un árbol de decisión que predice la salida en función de la entrada $\boldsymbol{x}$

Función Pérdida

La función de pérdida L mide la discrepancia entre la predicción del modelo $\hat{y}(x)$ y los valores verdaderos y . Para el conjunto de entrenamiento $\{(x_i, y_i)\}_{i=1}^{n}$, la función de pérdida total del modelo $\mathcal{L}$ se puede describir como:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} L(y_i, \hat{y}(x_i)) + \sum_{t=1}^{T} \Omega(f_t)$$

Donde:

- $\boldsymbol{L\left(y_i, \hat{y}(x_i)\right)}$ es la función de pérdida entre la predicción $\boldsymbol{\hat{y}(x_i)}$ y el valor real $\boldsymbol{y_i}$.

- $\boldsymbol{\Omega(f_t)}$ es el término de regularización que penaliza la complejidad del árbol $\boldsymbol{f_t}$. Este término ayuda a evitar el sobreajuste y se define típicamente como:

$$\Omega(f_t) = \gamma\, T_t + \frac{1}{2}\lambda \sum_{t=1}^{T} \omega_j^2$$

Donde:

- $\boldsymbol{T_t}$ es el número de nodos terminales del árbol $\boldsymbol{f_t}$.
- $\boldsymbol{\omega_j}$ es el peso de cada nodo terminal
- $\boldsymbol{\gamma}$ y $\boldsymbol{\lambda}$ son parámetros de regularización que controlan la complejidad del árbol.

Optimización de Segundo Orden

El algoritmo de XGBoost se basa en una aproximación de segundo orden para optimizar la función de pérdida en cada iteración del boosting. El término de optimización se realiza utilizando el gradiente y el hessiano de la función de pérdida. Para un modelo con T árboles, se aproximan los gradientes y hessianos de la función de pérdida en cada nodo del árbol.

Para un árbol f_t, el objetivo es minimizar la función de pérdida aproximada usando la expansión de Taylor de segundo orden:

$$\mathcal{L}(f_t) = \sum_{i=1}^{n} \left[g_i f_t\,(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t)$$

Donde:

- $\boldsymbol{g_i} = \frac{\boldsymbol{\partial L(y_i, \hat{y}(x_i))}}{\boldsymbol{\partial \hat{y}(x_i)}}$ es el gradiente de la función de pérdida.

- $\boldsymbol{h_i} = \frac{\partial^2 L(y_i, \hat{y}(x_i))}{\partial \hat{y}(x_i)^2}$ es el hessiano de la función de pérdida.

Iteración del Boosting

El proceso de entrenamiento de XGBoost consiste en agregar iterativamente árboles al modelo para corregir los errores cometidos por los árboles anteriores. En cada iteración t_i el árbol f_t se ajusta a los residuos del modelo previo utilizando los gradientes g_i y los hessianos h_i.

Los pasos para ajustar el modelo son los siguientes:

1. Inicializar las predicciones $\hat{y}_i{}^{(0)}$.
2. En cada iteración t_i, calcular los gradientes g_i y hessianos h_i en función de las predicciones actuales del modelo.
3. Construir un árbol f_t que minimice la función de pérdida aproximada usando gradientes y hessianos.
4. Actualizar las predicciones del modelo: $\hat{y}_i{}^{(t)} = \hat{y}_i{}^{(t-1)} + f_t(x_i)$
5. Repetir hasta alcanzar el número máximo de iteraciones o la convergencia.

1.2.2 Modelo CatBoost

CatBoost (Categorical Boosting) es un algoritmo de aprendizaje supervisado que utiliza la técnica de *gradient boosting*, construyendo modelos de manera secuencial, donde cada uno se enfoca en corregir los errores del modelo anterior. Su principal fortaleza radica en su capacidad de manejar variables categóricas de forma nativa, sin requerir codificación previa (como *one-hot encoding* o *label encoding*), gracias al uso de técnicas como *ordered target statistics* y *permutations*. Los *ordered target statistics* permiten calcular estadísticas de las variables categóricas basadas en la variable objetivo de manera ordenada, evitando el uso de información futura durante el entrenamiento, mientras que las *permutations* generan diferentes órdenes aleatorios de los datos para reducir la dependencia del modelo con el orden original, disminuyendo así el riesgo de *overfitting* y mejorando la calidad y estabilidad del modelo. La característica destacada de CatBoost es su uso de ordered boosting, una variante del gradient boosting tradicional que ayuda a evitar que el modelo utilice información del resultado que aún no debería conocer durante el entrenamiento, situación conocida como target leakage. Esto permite que el modelo aprenda de manera más realista y generalice mejor en datos nuevos. Además, maneja de forma eficiente los valores faltantes y utiliza una estructura de árboles simétricos optimizada, lo que mejora la velocidad de entrenamiento y la estabilidad del modelo. Con estas características, se destaca por su alto rendimiento en tareas de clasificación y regresión, especialmente en datasets con muchas variables categóricas, y se posiciona como una herramienta robusta, precisa y eficiente en problemas complejos del mundo real (Prokhorenkova et al., 2018).

Modelo de Predicción

El modelo de CatBoost se construye como una combinación secuencial de árboles de decisión. Para una entrada x la predicción $\hat{y}(x)$ de un modelo CatBoost con T árboles se expresa como:

$$\hat{y}(x) = \sum_{t=1}^{T} f_t\,(x)$$

Donde:

- $\boldsymbol{f_t(x)}$ es el $\boldsymbol{t}$ -ésimo árbol del modelo.
- La función $\boldsymbol{f_t(x)}$ es generalmente un árbol de decisión que predice la salida en función de la entrada $\boldsymbol{x}$

Función Pérdida con Regularización

La función de pérdida total a minimizar en CatBoost combina el error de predicción con un término de regularización para evitar el sobreajuste:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} L(y_i, \hat{y}(x_i)) + \sum_{t=1}^{T} \Omega(f_t)$$

Donde:

- $\boldsymbol{L\left(y_i, \hat{y}(x_i)\right)}$ es la función de pérdida entre la predicción $\boldsymbol{\hat{y}(x_i)}$ y el valor real $\boldsymbol{y_i}$.

- $\boldsymbol{\Omega(f_t)}$ es el término de regularización que penaliza la complejidad del árbol $\boldsymbol{f_t}$. Este término ayuda a evitar el sobreajuste y se define típicamente como:

$$\Omega(f_t) = \gamma\, T_t + \frac{1}{2}\lambda \sum_{t=1}^{T} \omega_j^2$$

Donde:

- $\boldsymbol{T_t}$ es el número de nodos terminales del árbol $\boldsymbol{f_t}$.
- $\boldsymbol{\omega_j}$ es el peso de cada nodo terminal
- $\boldsymbol{\gamma}$ y $\boldsymbol{\lambda}$ son parámetros de regularización que controlan la complejidad del árbol.

Ordered Boosting y Permutaciones

Una diferencia clave de CatBoost frente a otros algoritmos es el uso de Ordered Boosting, una técnica que evita el target leakage al entrenar cada modelo f_t con una estimación de gradientes obtenida a partir de una permutación aleatoria de los datos.

Se estima el gradiente para cada punto i como:

$$\boldsymbol{g_i} = \frac{\boldsymbol{\partial L(y_i, \hat{y}_{ordered}(x_i))}}{\boldsymbol{\partial \hat{y}}}$$

$$\boldsymbol{h_i} = \frac{\boldsymbol{\partial^2 L(y_i, \hat{y}_{ordered}(x_i))}}{\boldsymbol{\partial \hat{y}}}$$

Donde $\boldsymbol{\hat{y}_{ordered}(x_i)}$es la predicción que no incluye el punto al estimar el gradiente, evitando así el sesgo por uso de información futura.

Manejo de Variables Categóricas

CatBoost transforma variables categóricas usando estadísticas de objetivo ordenadas, como la media del target condicional a la categoría, con correcciones para evitar fugas de información. La codificación se construye de manera incremental:

$$\text{Encoded}\left(x_i{}^{(cat)}\right) = \frac{\sum_{j=1}^{i=1} 1(x_j{}^{(cat)} = x_i{}^{(cat)}).\, \boldsymbol{y_j} + \ a.P}{\sum_{j=1}^{i=1} 1(x_j{}^{(cat)} = x_i{}^{(cat)}) + a}$$

Donde:

- a es un parámetro de suavizado.
- P es la media global del target.
- Esta transformación se realiza con diferentes permutaciones del dataset.

Iteración del Boosting en Catboost

El proceso de entrenamiento sigue estos pasos:

1. Inicializar las predicciones $\hat{y}_i{}^{(0)}$.
2. En cada iteración t:
3. calcular los gradientes g_i y hessianos h_i con el ordered boosting.
4. Construir un árbol f_t que minimice la función de pérdida aproximada:

$$\sum_{i=1}^{n}[g_i f_t,(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \Omega(f_t)$$

5. Actualizar las predicciones del modelo: $\hat{y}_i{}^{(t)} = \hat{y}_i{}^{(t-1)} + f_t(x_i)$
6. Repetir hasta alcanzar el número máximo de iteraciones o la convergencia.

1.2.3 Evaluación de desempeño del Modelo

El desempeño de un modelo de regresión desarrollado con algoritmos de boosting se evalúa utilizando métricas que cuantifican la precisión de las predicciones generadas por el modelo en comparación con los valores reales observados (Chen & Guestrin, 2016; Prokhorenkova et al., 2018). Estas métricas permiten determinar qué tan bien el modelo generaliza sobre datos nuevos y cuantificar la magnitud de los errores de predicción.

Raíz del Error Cuadrático Medio (RMSE)

Es una métrica que indica la magnitud promedio de los errores cometidos por un modelo de predicción, midiendo la diferencia entre los valores predichos y los valores observados en las mismas unidades de la variable objetivo, lo que facilita su interpretación en la práctica (Shmueli et al., 2017). Esta métrica penaliza con mayor severidad los errores grandes debido a que se basa en el cuadrado de las diferencias antes de promediar y extraer la raíz cuadrada.

$$.\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\mathbf{2}}}$$

Donde:

- $\boldsymbol{y_i}$ es el valor observado o real para la $\boldsymbol{i}$ - ésima instancia.
- $\boldsymbol{\hat{y}_i}$ es el valor predicho para la $\boldsymbol{i}$ - ésima instancia.
- $\boldsymbol{n}$ es el número total de instancias.

Mean Absolute Error (MAE)

Es una métrica que calcula el promedio de los errores absolutos entre los valores predichos por un modelo y los valores observados, indicando en promedio cuánto se equivoca el modelo en sus predicciones (James et al., 2021). Al utilizar valores absolutos, cada error contribuye de forma uniforme a la métrica, lo que la hace menos sensible a valores atípicos y fácil de interpretar en la evaluación del desempeño del modelo.

Coeficiente de determinación ($\boldsymbol{R^2}$)

Es una métrica que evalúa qué tan bien un modelo explica la variabilidad de los datos observados, indicando la proporción de la varianza total de la variable objetivo que es explicada

por el modelo (Kuhn & Johnson, 2019). Un valor de R2 cercano a 1 indica que el modelo explica una alta proporción de la variabilidad de los datos, mientras que valores cercanos a 0 reflejan un bajo poder explicativo

1.2.4 Validación del Modelo

La validación en Machine Learning es el proceso utilizado para evaluar el desempeño de un modelo predictivo en datos no utilizados durante su entrenamiento, con el fin de estimar su capacidad de generalización y evitar el sobreajuste (overfitting) (James et al., 2021). Este paso es esencial para asegurar que el modelo no solo sea preciso en los datos que ha visto, sino que también mantenga un buen rendimiento cuando se aplique a datos nuevos en producción.

La validación permite estimar métricas de desempeño como RMSE, MAE y R^2 de forma confiable y detectar sobreajuste o subajuste en el modelo.

La validación mediante Bootstrap

Es un método de evaluación que utiliza muestreo aleatorio con reemplazo sobre el conjunto de datos disponible para estimar de forma robusta el error de predicción de un modelo (James et al., 2021). Este enfoque permite utilizar al máximo conjuntos de datos pequeños, generando múltiples muestras de entrenamiento y utilizando las observaciones no incluidas en cada muestra para evaluar el rendimiento del modelo. El procedimiento es el siguiente:

- Tomar B muestras Bootstrap del conjunto de datos original, cada una de tamaño igual al dataset, seleccionando con reemplazo.
- Entrenar el modelo en cada muestra Bootstrap.
- Evaluar el desempeño RMSE, MAE y R^2 del modelo en cada muestra.
- Promediar las métricas obtenidas para obtener una estimación estable del error de generalización del modelo.

2. Objetivos

2.1. Objetivo general

Estimar el ingreso de personas que no son clientes de la entidad financiera mediante la aplicación de algoritmos de boosting.

2.2. Objetivos específicos

- OE1. Examinar la calidad de los datos mediante un análisis exploratorio que permita identificar la distribución, consistencia y características de las variables continuas y categóricas.

- OE2. Aplicar técnicas de Machine Learning utilizando algoritmos de boosting, específicamente XGBoost y CatBoost, para la construcción del modelo de estimación de ingresos de personas que no son clientes de la entidad financiera.

- OE3. Evaluar y seleccionar el modelo con el mejor desempeño mediante métricas de precisión como RMSE, MSE y R^2, garantizando la efectividad y confiabilidad del estimador de ingresos propuesto.

- OE4. Validar la robustez y capacidad de generalización del modelo de estimación de ingresos mediante la técnica de bootstrap utilizando muestras distintas al conjunto de entrenamiento, para asegurar su consistencia y estabilidad en diferentes subconjuntos de datos.

3. Metodología del trabajo

3.1 Tipo de Investigación

La presente investigación adopta un enfoque cuantitativo de tipo explicativo, orientado al desarrollo de un modelo predictivo para la estimación de ingresos. Este enfoque se fundamenta en la necesidad de medir, procesar y analizar variables numéricas y categóricas provenientes de diversas fuentes de datos estructurados. Dicha aproximación permite la aplicación de técnicas estadísticas y de Machine Learning que aseguran resultados replicables, comparables y objetivos, esenciales para garantizar la rigurosidad y validez del análisis.

Asimismo, se trata de un estudio explicativo, ya que busca identificar y comprender relaciones causales o asociativas entre variables demográficas, de comportamiento crediticio y el nivel estimado de ingresos. El objetivo es explicar cómo interactúan dichos factores en el proceso de predicción, aportando así conocimiento útil para la toma de decisiones en contextos de inclusión financiera, análisis de riesgo y evaluación crediticia.

3.2 Diseño de Investigación

La presente investigación adopta un diseño no experimental, ya que no se realizará manipulación alguna sobre las variables independientes. En este tipo de diseño, los fenómenos se observan tal como se presentan en la realidad, sin intervención directa del investigador. Se parte de datos ya existentes, recolectados previamente por la entidad financiera u otras fuentes externas, lo cual permite trabajar con información real y confiable para el análisis. Esta elección metodológica resulta adecuada cuando el objetivo es estudiar relaciones entre variables en entornos naturales, sin necesidad de aplicar tratamientos o condiciones controladas.

El diseño no experimental permite desarrollar un modelo predictivo de estimación de ingresos para personas que no son clientes, utilizando variables demográficas, crediticias y de comportamiento observadas en registros históricos. Al no manipular dichas variables, sino

analizarlas tal como fueron registradas, se garantiza una representación fiel del comportamiento de los individuos evaluados. Este enfoque facilita la aplicación de técnicas estadísticas y de Machine Learning, asegurando un análisis riguroso que contribuya a una mejor toma de decisiones en procesos de evaluación crediticia.

3.3 Definición de la población

La población objetivo estará conformada por personas que laboran en el sector público en el Perú y que cuentan con ingresos registrados en el Portal de Transparencia del Estado, cumpliendo con el perfil definido por la entidad financiera. Se considerarán exclusivamente personas mayores de 30 años y menores de 69 años, en línea con las políticas de originación de préstamos de la entidad.

Para el desarrollo y validación del modelo, se considerará como ventana de observación (P.O.) el periodo comprendido entre enero de 2022 y mayo de 2025, utilizando estos datos para construir, entrenar y evaluar el desempeño del modelo estimador de ingresos. La ventana de construcción de variables abarcará una historia de hasta 12 meses previos al mes de referencia, utilizando julio de 2024 como mes de corte para la fase de desarrollo del modelo y mayo de 2025 como mes de corte para la fase de validación.

3.4 Fuentes de información

Para la construcción del modelo estimador de ingresos se utilizarán las siguientes fuentes de información externas:

- Padrón Electoral de RENIEC: Variables demográficas a nivel de número de documento.
- Reporte Crediticio Consolidado (RCC): Información de saldo crediticio y líneas de crédito.
- SUNEDU: Información de nivel educativo y grado académico.
- Inmuebles: Registro de propiedad de inmuebles.
- SUNAT: Información tributaria y actividad económica.
- Vehicular: Registro de propiedad vehicular.
- Portal de Transparencia: Registro de ingresos en el sector público.

3.5 Construcción de la variable respuesta

En cuanto a la definición de la variable respuesta (ingreso), durante la fase de desarrollo, que corresponde al periodo de enero de 2022 a julio de 2024, se identificará el último ingreso registrado de cada persona y se tomarán los últimos 12 meses en los que se hayan reportado ingresos, considerando únicamente aquellos casos en los que la persona cuente con al menos tres meses con ingresos superiores a S/ 1,130. A partir de estos registros, se calculará la mediana,

la cual será utilizada como el ingreso final para el entrenamiento del modelo. Para la fase de validación, que comprende el periodo de agosto de 2024 a mayo de 2025, se identificarán personas que no hayan sido incluidas en el conjunto de desarrollo y se seleccionará el último ingreso registrado superior a S/ 1,130 como el ingreso final que se utilizará en la validación del modelo.

3.6 Procedimientos de análisis de datos

3.6.1 Análisis univariado

Se realizará una reducción preliminar de variables con el objetivo de optimizar la calidad de los datos y asegurar la eficiencia del modelamiento. Esta reducción se efectuará mediante un análisis univariado, en el que se aplicarán los siguientes criterios de depuración:

- Completitud: Se calculará el porcentaje de valores faltantes de cada variable, eliminando aquellas que presenten más del 95 % de valores no informados, dado que su aporte al modelo sería marginal.
- Variabilidad: Se evaluará la desviación estándar de cada variable, conservando únicamente aquellas con una desviación estándar mayor a cero, con el fin de descartar variables sin variabilidad que no aporten información predictiva.
- Outliers: Se identificarán valores atípicos para cada variable continua y discreta, y se reemplazarán los valores que superen el percentil 99 % por el valor de dicho percentil, con el fin de evitar distorsiones en la distribución de las variables que puedan afectar la robustez del modelo.

Este proceso de filtrado inicial permitirá reducir el conjunto de variables a aquellas que posean información relevante, facilitando la posterior etapa de análisis multivariado y contribuyendo a la construcción de un modelo con mayor precisión y estabilidad.

3.6.2 Análisis multivariado

Se realizará un análisis de correlación con el propósito de identificar variables predictoras que presenten alta correlación entre sí, utilizando un umbral de 0.8 como criterio de detección. En los casos en que se identifique una correlación elevada, se conservará únicamente la variable que presente la mayor importancia media según los valores SHAP, con el fin de reducir la redundancia entre las variables explicativas y optimizar el rendimiento e interpretabilidad del modelo.

3.6.3 Partición muestral

- Se utilizará el 80% de la data para entrenamiento y el 20% para prueba.
- Validación fuera de tiempo (OOT): Se empleará un conjunto de datos de un periodo posterior al entrenamiento para simular la predicción en datos futuros no vistos y evaluar la estabilidad del modelo.

3.6.4 Construcción del modelo

En la construcción del modelo, se empleará CatBoost utilizando directamente las variables categóricas en su estructura original, permitiendo integrarlas al entrenamiento sin transformaciones adicionales. En paralelo, para XGBoost se transformarán las variables categóricas en variables dicotómicas (flags) antes de su inclusión, asegurando la adecuada lectura por el algoritmo durante el modelamiento. Este enfoque permitirá comparar el desempeño de ambos algoritmos bajo el mismo conjunto de datos y evaluar cuál proporciona una mejor estimación del ingreso para personas que no son clientes de la entidad.

El ajuste de hiperparámetros se realizará utilizando la librería Optuna, configurada para ejecutar búsquedas automáticas y sistemáticas que permitan identificar las combinaciones de parámetros que maximizan el desempeño predictivo de los modelos XGBoost y CatBoost. Este ajuste se aplicará sobre el conjunto de entrenamiento, utilizando las métricas de desempeño definidas.

Posteriormente, se evaluará la importancia de las variables mediante la media de los valores SHAP obtenidos en el conjunto de entrenamiento, con el propósito de identificar aquellas variables que generan mayor aporte predictivo al modelo. Esta evaluación permitirá realizar ajustes en la selección de variables, facilitar la interpretación de los resultados y contribuir a la transparencia y robustez del modelo.

3.6.5 Validación y métricas de desempeño

La validación del modelo se realizará tras finalizar el entrenamiento utilizando la muestra de prueba y la muestra de validación fuera de tiempo, con el fin de evaluar el desempeño predictivo en datos no utilizados durante la etapa de construcción. En esta etapa, se aplicará bootstrap para obtener estimaciones de las métricas de desempeño. Se emplearán el RMSE y el MSE para cuantificar la magnitud del error en las predicciones de ingresos, mientras que el R^2 se utilizará para evaluar la proporción de la variabilidad del ingreso explicada por el modelo. Estas métricas se calcularán tanto en la muestra de prueba como en la muestra de validación fuera de tiempo, permitiendo medir la capacidad del modelo.

3.7 Herramientas de análisis de datos

Para el desarrollo del modelo, se utilizará Python como lenguaje de programación debido a su versatilidad y eficiencia en el manejo de grandes volúmenes de datos y la implementación de algoritmos de machine learning. El entorno de trabajo seleccionado será Jupyter Notebook, ya que permite una integración práctica de código, visualizaciones durante el proceso de construcción y validación del modelo.

Se emplearán las siguientes librerías de Python:

- pandas y numpy, utilizadas para la manipulación, limpieza y transformación de datos, facilitando la construcción de variables y el manejo de estructuras de datos eficientes.
- scikit-learn, para la partición de datos en conjuntos de entrenamiento y prueba, el escalado de variables y el cálculo de métricas de desempeño del modelo.

- xgboost y catboost, para la implementación de algoritmos de boosting que permitan modelar relaciones no lineales y manejar adecuadamente variables categóricas durante el entrenamiento del modelo de predicción de ingresos.
- matplotlib y seaborn, para la generación de gráficos de análisis exploratorio de datos y visualización de resultados del modelo, facilitando la interpretación de los patrones encontrados en las variables.

4. Cronograma

Actividad/semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Elaboración del Proyecto de TSP																
Aprobación del Proyecto TSP																
Recopilación y sistematización de los datos																
Proceso analítico de los datos																
Redacción preliminar del TSP																
Generación del borrador completo del TSP																
Revisión y corrección del TSP																
Aprobación y disertación del TSP																

5. Referencias bibliográficas

Altman, E. I., Iwanicz-Drozdowska, M., Laitinen, E. K., & Suvas, A. (2018). *Financial distress prediction in an international context: A review and empirical analysis of Altman's Z-score model*. *Journal of International Financial Management & Accounting, 29*(2), 131–171. https://doi.org/10.1111/jifm.12053

Berg, T., Burg, V., Gombović, A., & Puri, M. (2020). *On the rise of FinTechs: Credit scoring using digital footprints*. *The Review of Financial Studies, 33*(7), 2845–2897.

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://www.statlearning.com/

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). *CatBoost: Unbiased boosting with categorical features*. *Advances in Neural Information Processing Systems, 31*, 6638–6648. https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html

Superintendencia de Banca, Seguros y AFP. (2023). *Resolución S.B.S. N° 02738-2023: Reglamento para la evaluación y clasificación del deudor y la exigencia de provisiones*.

Superintendencia de Banca, Seguros y AFP del Perú. https://www.sbs.gob.pe/app/Normas/Leyes/ResSBSPDF/02738-2023.pdf

Thomas, L. C., Crook, J. N., & Edelman, D. B. (2017). *Credit scoring and its applications* (2nd ed.). SIAM. https://doi.org/10.1137/1.9781611974555

World Bank. (2024). *The use of alternative data in credit risk assessment: Opportunities, risks, and challenges*.