Predicción de gastos personales o familiares de los clientes de Bancolombia

David Alberto Rodriguez Muñoz

Monografía presentada como requisito parcial para optar al título de: **Especialización en analítica y ciencia de datos**

Tutor

Efraín Alberto Oviedo Carrascal

Universidad de Antioquia

Facultad de Ingeniería

Especialización en analítica y ciencia de datos

Medellín, Antioquia

2021

Cita	(Rodriguez Muñoz, 2021)
Referencia **Estilo APA 7 (2020)**	Rodriguez Muñoz, D. A. (2021). *Predicción de gastos personales o familiares de los clientes de Bancolombia* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín

Especialización en analítica y ciencia de datos

Cohorte II

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego Jose Luis Botia Valderrama

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. RESUMEN EJECUTIVO

Para los bancos es de vital importancia poder anticiparse a la realidad económica de cada persona (y más cuando el mismo es cliente del propio banco), para así determinar planes de acción con ese cliente, planes que incluyen: ofertar créditos hipotecarios, créditos de consumo o tarjetas de crédito, entre muchos otros productos que pueda ofrecer el banco. Este es el objetivo principal de esta monografía, hacer uso de un conjunto de datos demográficos y financieros de los clientes del banco Bancolombia, para diseñar un estimador basado en algoritmos de aprendizaje automático, que tengan la capacidad de adelantarse con un alto grado de predicción a los gastos personales que tendrá el cliente del banco en los próximos meses, con el fin de ayudar al banco en la creación de créditos y el ajuste de la capacidad de pago de cada cliente. Los datos con los que se cuenta para el diseño del algoritmo, son anonimizados, obtenidos mediante la plataforma kaggle, donde los mismos fueron publicados para la competencia Dataton BC 2020, estos datos describen: la vida financiera de cada cliente, si presenta algún tipo de crédito con el banco, las obligaciones financieras que pueda tener, si ha tenido cartera castigada o si el cliente ha estado mucho tiempo en mora, entre otros (además de datos demográficos). Las estrategias utilizadas para solucionar el problema mediante la creación del diseño más óptimo posible fueron: trabajar fuertemente sobre diferentes transformaciones de los datos, adicionar y eliminar varios datos o utilizarlos de una manera diferente (muestras del conjunto inicial), realizar feature engineering para crear características que permitan la disminución de la dimensionalidad del conjunto de datos. Durante cada tratamiento sobre el conjunto de datos, se utilizan diseños sencillos de algoritmos de aprendizaje de máquina para analizar los efectos que estos cambios tienen sobre el modelo en cuestión. El tratamiento de los datos no fue sencillo, se encontraron las siguientes observaciones; hay diferentes valores nulos sobre los datos (tanto numéricos como categóricos), la dispersión de los datos numéricos fue un tema importante, ya que hubo características cuya naturaleza no es propia para todos los individuos del banco (si no para un sector particular), los datos se encontraban desactualizados, analizando las variables categóricas se encuentra la existencia de un sesgo poblacional fuerte, por último, las características no presentaban la correlación necesaria para describir la variable objetivo. De todos los modelos diseñados, el mejor resultado obtenido fue el Gradient Boost Tree con una profundidad máxima de 15 y 50 árboles estimadores (rendimiento en R2 de 13.8% para entrenamiento y 12.4% para prueba, y, MAPE 111 para entrenamiento y 109 para prueba), cabe resaltar que es el modelo con mejores resultados al evaluar con los datos de prueba, existen otros modelos que logran un rendimiento más alto en el conjunto de entrenamiento, pero disminuye en el conjunto de prueba. Estos resultados podrían mejorarse aplicando las sugerencias presentadas para trabajos futuros.

https://github.com/Dave0995/monografia

2. DESCRIPCIÓN DEL PROBLEMA

Conocer la realidad económica de los clientes del banco es importante para promover planes de trabajo, iniciativas y desarrollos que permitan impulsar la economía del país. El objetivo es realizar una predicción de los gastos personales y/o familiares de los clientes del banco, de esta manera se facilitan los estudios crediticios y se ajustan las capacidades de pago de los individuos para que una mayor cantidad de clientes se vean beneficiados con el acceso a los préstamos. Este problema se puede resolver utilizando aprendizaje automático supervisado, haciendo uso de una muestra aleatoria del conjunto de datos se puede realizar análisis exploratorio y feature engineering, con lo cual se transforman las características del conjunto de datos inicial, para que sea fácilmente interpretable por un modelo de regresión, el cual a través de las relaciones entre las características y la variable objetivo, aprenderá la mejor manera de calcular la predicción.

2.1 PROBLEMA DE NEGOCIO

La empresa Bancolombia se encuentra interesada en tener conocimiento preventivo sobre la realidad económica de sus clientes, saber cuáles son sus gastos mensuales le permitirá ajustar las capacidades de pago y endeudamiento que tiene cada individuo, de esta manera, ofrecer de manera oportuna el producto que más se ajuste a las necesidades crediticias de cada persona. Para lograr este objetivo ha propuesto una competencia a nivel nacional en la plataforma kaggle, donde provee a los competidores con datos anonimizados de los clientes del banco.

2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS

Se desarrollarán modelos que permitan al banco obtener un estimado de los gastos familiares de sus antigüos y nuevos clientes para poder monitorear sus necesidades crediticias, ofreciendo de manera rápida los productos que tiene en su portafolio y seleccionando el más adecuado a la necesidad financiera del cliente. De esta manera, más clientes podrán acceder a productos crediticios, esto se convierte en ganancias por conceptos de cuotas de manejo y tasas de interés para el banco. Además, al tener con precisión la realidad económica del cliente, se evitarían pérdidas al sobreestimar la capacidad de endeudamiento de los clientes y se aumentan las ganancias al no subestimar las capacidades de endeudamiento de otros.

2.3 ORIGEN DE LOS DATOS

Se cuenta con datos demográficos y financieros de un aproximado de 20 millones de clientes que tiene el banco a nivel nacional. Estos datos son históricos y algunos se encuentran desactualizados, con ellos se puede describir el comportamiento financiero de los clientes, si no tienen créditos activos, que tipos de créditos ha solicitado, a qué grupo de riesgo pertenece, a qué

grupo de segmentación pertenece, su ciudad de trabajo, nivel académico, etc. Es una muestra aleatoria de clientes del banco en sus diferentes localidades.

2.4 MÉTRICAS DE DESEMPEÑO

Las métricas consideradas para este trabajo son las siguientes:

Métricas de machine learning

- **Coeficiente de determinación (R2).** Esta métrica nos permite conocer en base a una correlación, que tan buenas son las predicciones que realizan los modelos diseñados. Eso lo hace teniendo en cuenta la variación que existe entre el valor real y el valor predicho [1]. El valor objetivo de esta métrica se encuentra por encima del 0.9, con el fin de confirmar que tan buen ajuste tiene el modelo con respecto a los datos nuevos.

- **Mean absolute percentage Error (MAPE).** Es una métrica de desempeño para problemas de regresión la cual mide el tamaño del error en términos porcentuales [2]. Dada la naturaleza del problema, es importante tomar como base un MAPE del 5%, es decir, al realizar una predicción con el modelo, en promedio, el pronóstico está errado en un 5%.

El objetivo de usar el MAPE es porque esta métrica es la requerida por la competencia, pero adicional a ello, es buena para obtener una idea de que tan errada se encuentra una predicción del valor real. El inconveniente es que está métrica es demasiado sensible a la partición de los datos al momento de evaluar un modelo, para ello, se implementó el R2, como una métrica auxiliar que permitiera corroborar los resultados obtenidos con el MAPE. De esta manera, se tienen dos métricas que juntas permiten tener un panorama más claro del rendimiento de un modelo, una indica el error en la predicción y otra indica que tan lejos se encuentra la predicción del ajuste realizado por el modelo durante el entrenamiento.

Métricas de negocio

Dado que la problemática proviene de una competencia de kaggle, no existe un antecedente que permita comparar los resultados del modelo diseñado, en algún tipo de valor tangible para el negocio. Por lo anterior, solo se realizará la propuesta de una métrica de negocio que pueda funcionar como antecedente para predicciones futuras.

La necesidad del banco para conocer de manera preventiva la realidad económica de sus clientes, es para poder determinar si un cliente es buen candidato para la oferta de algún tipo de crédito.

Como métrica de negocio se propone la cantidad de clientes que acceden a un nuevo régimen crediticio por un periodo de un mes. Basándose en los resultados obtenidos por el modelo, el personal del banco tendrá acceso a información que le permitirá seleccionar de mejor manera el crédito más conveniente para un cliente. Por otro lado, ya dependerá del cliente si se acepta o no el crédito ofertado por el banco, y en base al número de créditos generados, se evaluará la importancia que tiene el uso del modelo en el negocio.

3. DATOS

3.1 DATOS ORIGINALES

Los datos de los que dispone la competencia son un archivo comprimido en formato zip, que contiene un archivo en formato csv y separado por ";". Las columnas de este dataset contienen información demográfica y financiera de los clientes del banco (completamente anonimizada).

La tabla 1 contiene los Metadatos del conjunto de datos del proyecto, en la cual se puede encontrar la siguiente información; el nombre de las variables, su tipo y una breve descripción de qué significa cada una.

Variable	Tipo	Descripción
periodo	int	Periodo de estimación para el cliente
id_cli	bigint	Número de identificación del cliente
fecha_nacimiento	bigint	Fecha de nacimiento del cliente
edad	double	Edad del cliente
género	string	Género reportado por el cliente
estado_civil	string	Estado civil reportado por el cliente
nivel_academico	string	Nivel académico reportado por el cliente
profesion	string	Profesión reportado por el cliente
ocupacion	string	Ocupación reportado por el cliente
tipo_vivienda	string	Tipo vivienda reportado por el cliente
ult_actual	bigint	Fecha en la cual se actualizó la información del cliente por última vez
categoría	tinyint	Categorización de los clientes según sus ingresos. Es el segmento del cliente. Los clientes se encuentran separados por segmentos al interior del banco. Por motivos de confidencialidad la segmentación está oculta pues es una segmentación propia del negocio
codigo_ciiu	bigint	Actividad económica según clasificación de la DIAN
ind_mora_vigente	string	Indica si el cliente se encuentra en mora

cartera_castigada	string	Indica si el cliente tiene cartera castigada al momento
ciudad_residencia	string	Ciudad de residencia reportado por el cliente
departamento_residencia	string	Departamento de residencia reportado por el cliente
ciudad_laboral	string	Ciudad laboral reportado por el cliente
departamento_laboral	string	Departamento laboral reportado por el cliente
rechazo_credito	string	Flag que indica rechazo de crédito en el banco
mora_max	bigint	Número máximo de días en mora que ha tenido el cliente
cant_moras_30_ult_12_meses	bigint	Cantidad de moras superiores a 30 días en los últimos 12 meses
cant_moras_60_ult_12_meses	bigint	Cantidad de moras superiores a 60 días en los últimos 12 meses
cant_moras_90_ult_12_meses	bigint	Cantidad de moras superiores a 90 días en los últimos 12 meses
cupo_total_tc	double	Cupo total en tarjetas de crédito. Es la sumatoria de los cupos en tdc en Bancolombia y competencia
tenencia_tc	string	Flag que indica tenencia de tarjetas de crédito
cuota_tc_bancolombia	double	Cuota paga de tarjeta de crédito en el banco
tiene_consumo	string	Flag de tenencia crédito de consumo
tiene_crediagil	string	Flag de tenencia Crédiágil
nro_tot_cuentas	bigint	Número total de cuentas del cliente. (Cuentas de Ahorro y Corriente)
ctas_activas	bigint	Cuentas activas del cliente
tiene_ctas_activas	string	Flag de tenencia de cuentas activas
ctas_embargadas	bigint	Número de cuentas embargadas
tiene_ctas_embargadas	string	Flag de tenencia de cuentas embargadas
pension_fopep	string	Es pensionado por FOPEP
cuota_cred_hipot	double	Cuota del crédito hipotecario en el banco
tiene_cred_hipo_1	string	Tiene crédito hipotecario tipología 1
tiene_cred_hipo_2	string	Tiene crédito hipotecario tipología 2
mediana_nom3	double	Mediana últimos 3 meses del ingreso por nómina
mediana_pen3	double	Mediana últimos 3 meses del ingreso por pensión
ingreso_nompen	double	Ingreso por concepto de nómina, pensión, pago a proveedores
cat_ingreso	string	Categoría del ingreso, nómina, pensión
ingreso_final	double	Ingreso final del cliente

cant_mora_30_tdc_ult_3m_sf	double	Cantidad de moras de 30 días en tarjeta de crédito durante los últimos 3 meses en el sector financiero
cant_mora_30_consum_ult_3m_sf	double	Cantidad de moras de 30 días en crédito de consumo durante los últimos 3 meses en el sector financiero
cuota_de_vivienda	double	Cuota créditos de vivienda en el sector financiero
cuota_de_consumo	double	Cuota créditos de consumo en el sector financiero
cuota_rotativos	double	Cuota créditos rotativos en el sector financiero
cuota_tarjeta_de_credito	double	Cuota total a pagar tarjeta de crédito
cuota_de_sector_solidario	double	Cuota total a pagar de sector solidario
cuota_sector_real_comercio	double	Cuota total a pagar del sector real
cupo_tc_mdo	double	Cupo total de tarjeta de crédito en el mercado (por fuera de Bancolombia)
saldo_prom3_tdc_mdo	double	Saldo promedio de los últimos 3 meses en tarjetas de crédito
cuota_tc_mdo	double	Cuota a pagar en tarjetas de crédito en el mercado
saldo_no_rot_mdo	double	Saldo adeuda en créditos no rotativos en el mercado
cuota_libranza_sf	double	Cuota a pagar en libranza en el sector financiero
cant_oblig_tot_sf	double	Cantidad de obligaciones totales del sector financiero
cant_cast_ult_12m_sr	double	Cantidad de carteras castigadas en los últimos 12 meses
ind	double	Ingreso neto disponible calculado para el cliente Ingreso final – Gasto familiar – Cuotas pagadas + Cuotas pagadas de la línea de crédito Libranza. Cuotas Pagadas = Cuota de vivienda (CUOTA DE VIVIENDA) + cuota de consumo (CUOTA DE CONSUMO) + cuota rotativos (CUOTA ROTATIVOS) + cuota comercial (CUOTA COMERCIAL) + cuota de microcrédito (CUOTA DE MICROCRÉDITO) + cuota de TDC (CUOTA TARJETA DE CREDITO) + cuota de sector solidario (CUOTA DE SECTOR SOLIDARIO) + cuota sector real comercio (CUOTA SECTOR REAL COMERCIO).
rep_calif_cred	string	Grupo de riesgo
pol_centr_ext	double	Flag de cumplimiento de políticas de riesgo central externa
convenio_lib	string	Flag que indica convenio de libranza con la empresa que labora

ingreso_nomina	double	ingreso reportado por pago de nómina
ingreso_segurida_social	bigint	Ingreso reportado por pagos a seguridad social
gasto_familiar	double	Gasto familiar del cliente, esta es la variable a estimar

Tabla 1. Metadatos del proyecto.

En total se cuenta con 20.988.748 registros anonimizados de los clientes del banco. Estos datos en su totalidad pesan aproximadamente 9 Gb y no presentan ningún tipo de restricción para su uso (sólo es necesario registrarse en la competencia para obtener una copia de la información). La variable a estimar es el gasto familiar.

3.2 DATASETS

Para la construcción del conjunto de datos es necesario trabajar todo el dataset con Spark, así es posible manejar ese gran volumen de información en un entorno local. Para cargar los datos, fue necesario realizar una definición del esquema del conjunto de datos, en este proceso es importante la tabla 1 donde se especifican tanto nombres como tipos de las variables. Por otro lado, para optimización de procesamiento, y para aumentar los tiempos de carga y transformación, se convierten los datos a formato parquet.
Utilizando el conjunto de datos en formato parquet se crea una muestra aleatoria del conjunto de datos original, esta muestra corresponde al 2.5% de los datos totales (524.623 registros). De esta manera, el conjunto de datos se hace más manejable y de fácil trabajo en una instancia de baja capacidad (cpu de 8 cores y 8gb de Ram).

Para hacer la validación de los modelos, la muestra poblacional se divide en dos partes a una relación 70/30 para los datos de entrenamiento y prueba. Pero, viendo la necesidad de tener un mejor panorama sobre el rendimiento de los modelos, y al notar la dificultad que representan los datos, se cambió esta metodología de evaluación a una por validación cruzada con cinco grupos de datos.

3.3 DESCRIPTIVA

La siguiente tabla representa cinco registros de todo el conjunto de datos. Los ejes fueron invertidos con el fin de obtener una mejor representación de los datos en un mismo eje y así evitar mostrar particiones de esas cinco muestras del conjunto de datos.

periodo	202003	201902	202001	202008	202001
id_cli	2089776	2088089	3892351	2897552	4782141

fecha_nacimiento	19840630	19860727	19910108	19900903	19790623
edad	356,386,037	3,247,638,604	2,893,634,497	2,988,364,134	4,048,186,174
género	M	M	M	M	F
estado_civil	DIVORCIADO	UNIÓN LIBRE	SOLTERO	SOLTERO	NO INFORMA
nivel_academico	TECNÓLOGO	NO INFORMA	TECNÓLOGO	BACHILLER	SIN INFORMACIÓN
profesion	TECNOLOGIA SISTEMAS	\N	OTROS	\N	\N
ocupacion	Empleado	Independiente	Independiente	Empleado	Empleado
tipo_vivienda	ALQUILADA	FAMILIAR	\N	\N	\N
ult_actual	20180526	20181120	20190802	20190906	20191211
categoría	1	4	4	1	1
codigo_ciiu	10	8230	10	10	10
ind_mora_vigente	N	N	N	N	N
cartera_castigada	N	N	N	N	N
ciudad_residencia	CALI	PALMIRA	MEDELLIN	MEDELLIN	BOGOTA D.C.
departamento_residencia	VALLE	VALLE	ANTIOQUIA	ANTIOQUIA	CUNDINAMARCA
ciudad_laboral	CALI	\N	\N	MEDELLIN	\N
departamento_laboral	VALLE	\N	\N	ANTIOQUIA	\N
rechazo_credito	\N	\N	\N	\N	\N
mora_max			0	1	0
cant_moras_30_ult_12_meses			0	0	0

cant_moras_60_ult_12_meses			0	0	0
cant_moras_90_ult_12_meses			0	0	0
cupo_total_tc	0	0	0	0	0
tenencia_tc	NO	NO	SI	NO	NO
cuota_tc_bancolombia	0	0	0	0	0
tiene_consumo	\N	\N	\N	\N	X
tiene_crediagil	\N	\N	\N	\N	\N
nro_tot_cuentas	1	1	1	1	1
ctas_activas	1	1	1	1	1
tiene_ctas_activas	X	X	X	X	X
ctas_embargadas	0	0	0	0	0
tiene_ctas_embargadas	\N	\N	\N	\N	\N
pension_fopep	\N	\N	\N	\N	\N
cuota_cred_hipot					
tiene_cred_hipo_1	\N	\N	\N	\N	\N
tiene_cred_hipo_2	\N	\N	\N	\N	\N
mediana_nom3	1255032	0	0	0	4353538
mediana_pen3	0	0	0	0	0
ingreso_nompen	1255032	0	0	0	4353538
cat_ingreso	NOM	\N	\N	\N	NOM
ingreso_final	1391032	2327500	6519750	1484205	4353334
cant_mora_30_tdc_ult_3m_sf	0			0	0
cant_mora_30_consum_ult_3m_sf	0			0	0
cuota_de_vivienda	0	0	0	0	0
cuota_de_consumo	0	0	0	0	386578
cuota_rotativos	0	0	0	0	11000

cuota_tarjeta_de_credito	0	0	0	0	1006000
cuota_de_sector_solidario	0	0	0	0	0
cuota_sector_real_comercio	0	0	0	524000	28000
cupo_tc_mdo	0	0	0	0	2,58E+07
saldo_prom3_tdc_mdo	0	0	0	0	1,62E+07
cuota_tc_mdo	0	0	0	0	751000
saldo_no_rot_mdo	0	0	0	2555000	211000
cuota_libranza_sf	0	0	0	0	0
cant_oblig_tot_sf	0			0	4
cant_cast_ult_12m_sr	0			1	0
ind	695516	1187025	3879251,25	210,681,475	1615755,8
rep_calif_cred	C	SIN INFO	SIN INFO	F	C
pol_centr_ext	0			7	0
convenio_lib	\N	\N	\N	\N	70831
ingreso_nomina	1255032				4353538
ingreso_segurida_social				1484205	3500000
gasto_familiar	304687	187990	862348,92	1056864	248386

Tabla 2. Ejemplo de cinco registros de los datos del proyecto.

El análisis exploratorio de este conjunto de datos se realizó bajo las siguientes preguntas:

- ¿En cuáles departamentos se presenta la mayor concentración de clientes del banco?
- En base al nivel académico ¿Quienes presentan mayor ingreso neto disponible teniendo en cuenta si tienen algún tipo de crédito?
- ¿Existe algún sesgo en los datos, es decir, al momento de realizar el dataset, las variables categóricas presentan buen balance o los datos demuestran que existe una tendencia particular hacia un grupo de la sociedad?

- Teniendo en cuenta la anterior pregunta ¿Se puede definir a los clientes del banco?

La figura 1 nos permite responder la pregunta **¿En cuáles departamentos se encuentra la mayor concentración de clientes del banco?** La respuesta deja una preocupación notable, los clientes del banco son principalmente personas que se encuentran en los departamentos de Antioquia y Cundinamarca (BOGOTÁ D.C.), de la totalidad de los datos un 54.4% corresponde a los clientes de los departamentos mencionados. Ahora bien, la preocupación proviene del hecho de que Antioquia y Cundinamarca hacen parte de los departamentos con más desarrollo económico del país, es decir, al ingresar los datos en el modelo existirá una tendencia hacia los valores económicos de estos departamentos.

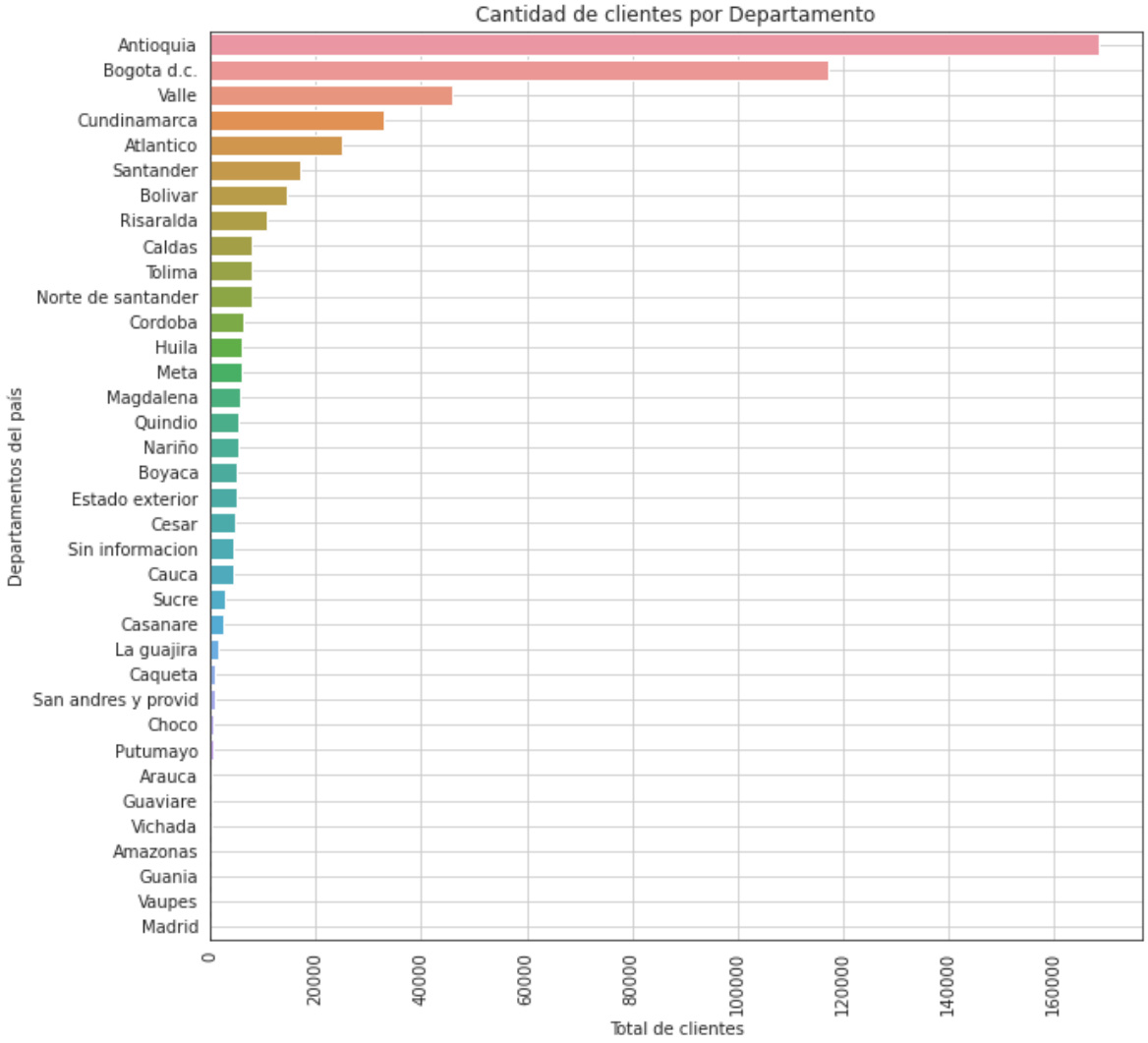

Figura 1. Clientes totales por cada departamento.

La segunda pregunta se puede responder utilizando como base la figura 2, ya que nos da información interesante sobre las personas con mayores ingresos (a priori). Bajo la creencia popular de que estudiar y hacer un postgrado nos ayuda a impulsar nuestros ingresos, podemos

afirmar bajo los datos provistos por el banco, que los clientes con mayor ingreso neto disponible son aquellos que solo han culminado la educación básica primaria y que por el contrario, los clientes con ingresos netos por debajo de 2 millones, son los universitarios y especialistas. Si acompañamos la gráfica anterior con la tabla 3, observamos que las personas sin formación académica, son las menos interesadas en adquirir créditos bancarios, mientras que, las personas con educación profesional son quienes más créditos solicitan. De lo anterior nace otra pregunta, ¿Qué tan representativa es esta información si tenemos en cuenta la cantidad de personas por cada categoría de la variable nivel académico? Al analizar la figura 3, se concluye que es poco representativa esta información, dado que nuevamente el conjunto de datos se encuentra sesgado hacia la categoría universitario con aproximadamente 200 mil de los datos (sin mencionar las categorías nulas donde el cliente no informa su nivel académico). Por lo tanto, toda esta información anidada, permite concluir que este dataset presenta graves problemas a nivel de distribución, balance y variabilidad de datos.

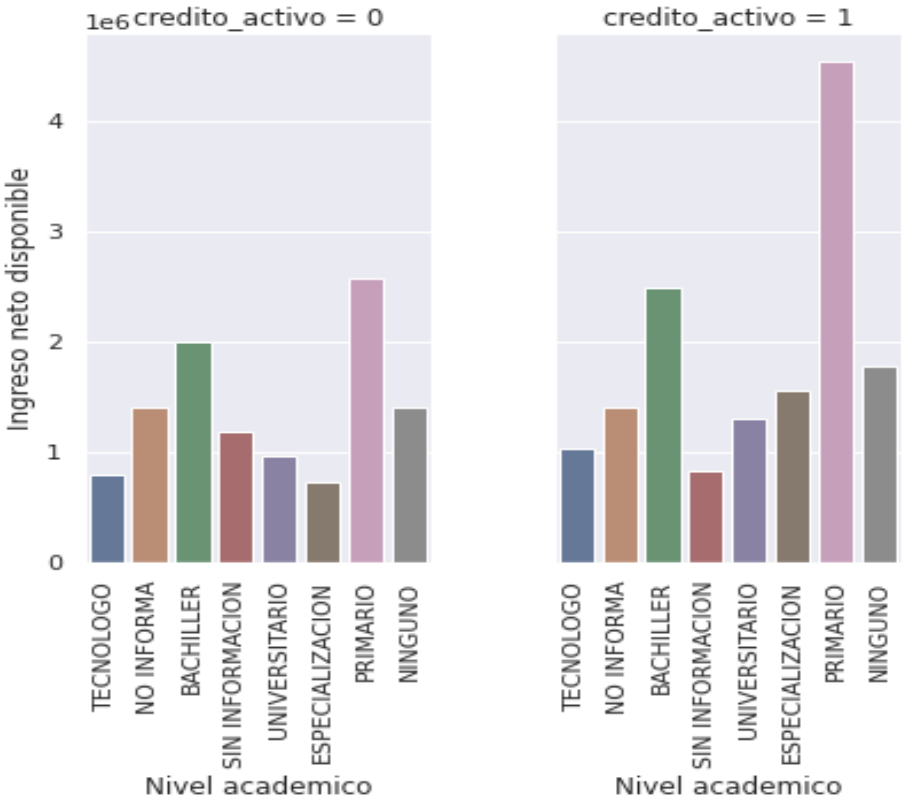

Figura 2. Gráficas de Ingreso neto disponible para cada cliente con diferente nivel académico y si tiene o no un crédito activo con el banco.

NIVEL ACADÉMICO	CRÉDITO ACTIVO

NINGUNO	1860
PRIMARIO	2386
SIN INFORMACIÓN	17794
ESPECIALIZACIÓN	20568
BACHILLER	22058
NO INFORMA	27726
TECNÓLOGO	30479
UNIVERSITARIO	82171

Tabla 3. Total de créditos activos en cada categoría del nivel académico.

Figura 3. Total de clientes por categoría de nivel académico.

Finalmente, utilizando las respuestas de las preguntas anteriores y apoyándonos de las gráficas del tipo de vivienda (figura 4), estado civil y la ocupación (figuras 5 y 6), podemos concluir de manera definitiva que existe un sesgo en los datos. Además, podemos realizar una descripción promedio de los clientes del banco (basados por supuesto en los datos compartidos en la competencia). Las figuras 4, 5 y 6, evidencian que las variables categóricas del conjunto de datos están completamente centradas en un público en particular, ese público son: personas de nivel universitario, solteras que actualmente cuentan con una relación laboral estable con una empresa (ya sea por contrato a término definido o contrato a término indefinido), viven con sus padres, no

se les ha negado un crédito y tampoco es que los soliciten. Teniendo este perfil de los clientes del banco es posible lanzar campañas con créditos hipotecarios o créditos de vivienda, por ejemplo.

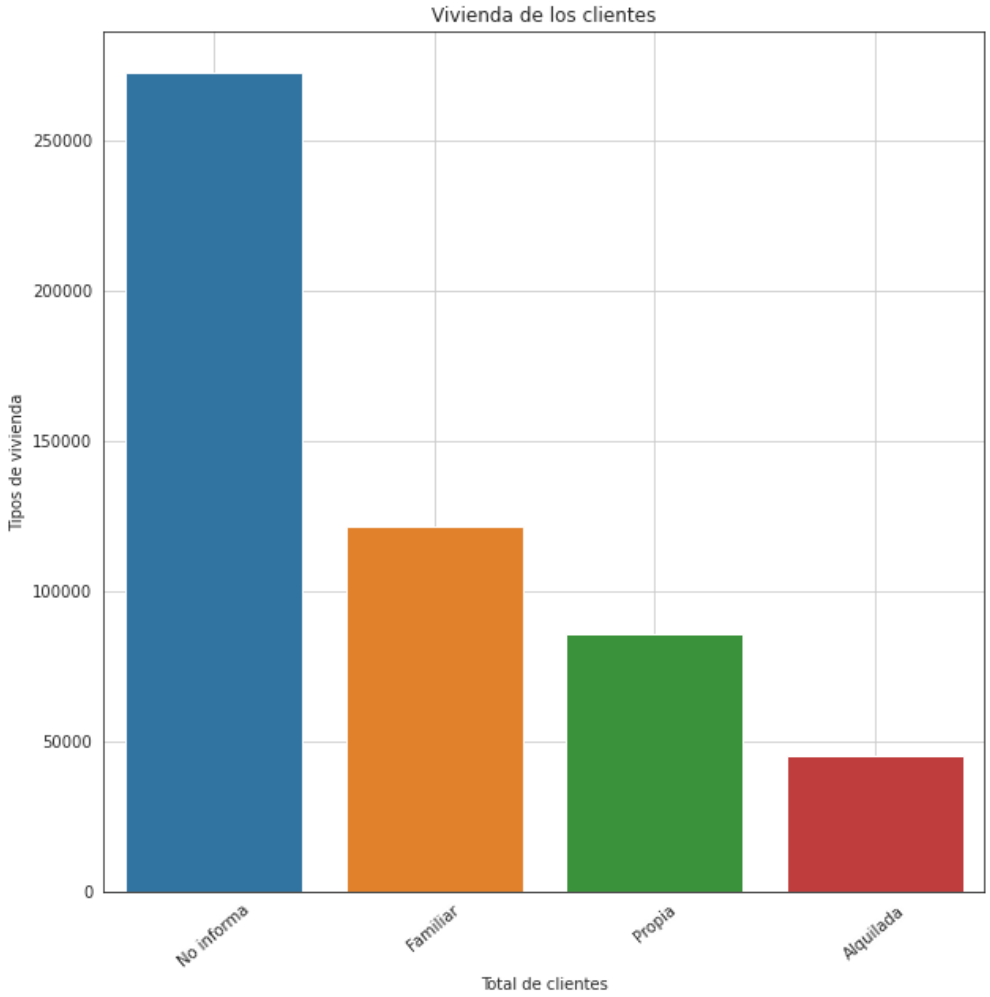

Figura 4. Gráfico de tipo de vivienda para los clientes.

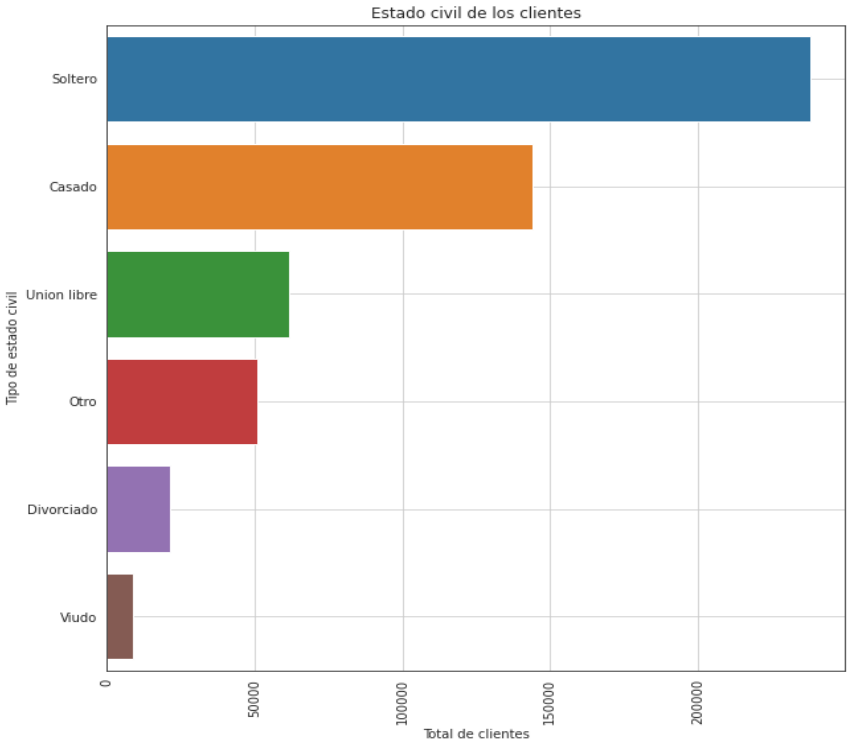

Figura 5. Gráfico de estado civil para los clientes.

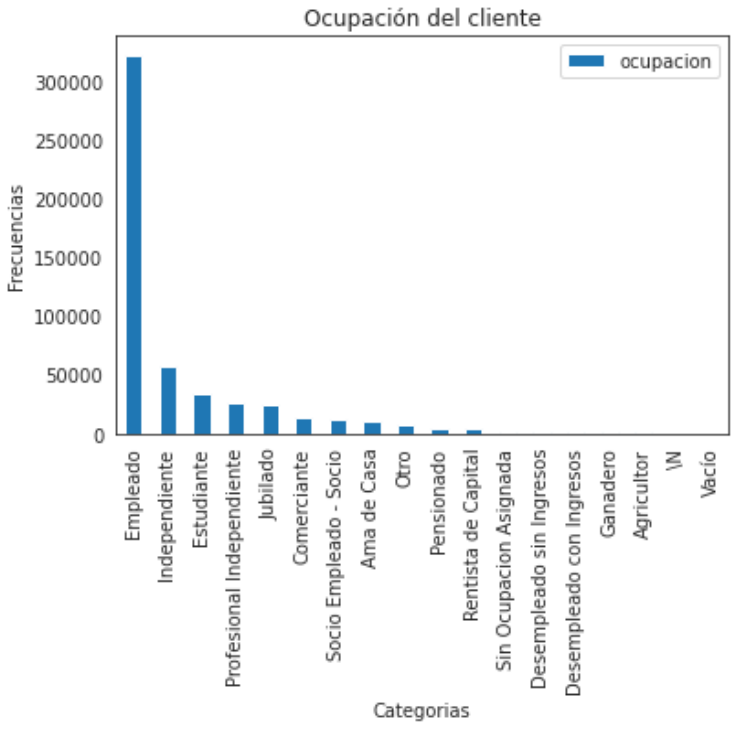

Figura 6. Gráfico de la ocupación de los clientes.

Por lejos, la figura 6 es la más interesante. Esta gráfica indica la centralización de los datos en una población particular, la muestra poblacional es de 524.623 clientes, de los cuales aproximadamente el 60% son personas que se encuentran en una relación laboral con un contrato a término definido o indefinido. Lo mencionado anteriormente, es un sesgo importante, dado que la mayoría de las personas en Colombia, no se encuentran en una relación laboral de este tipo, existen muchos empleados por contrato de prestación de servicios o por cuentas de cobro (como consultores o especialistas). Estos sesgos son importantes porque de cierta manera definen la tendencia de predicciones de los modelos, ya que sus predicciones se constituyen en los datos que lo definen y con el cual fueron entrenados.

4. PROCESO DE ANALÍTICA

4.1 PIPELINE PRINCIPAL

El diagrama en la figura 7 define el pipeline del trabajo desarrollado. Consta de múltiples etapas lineales y finaliza con la etapa iterativa, donde se prueban las diferentes configuraciones de los modelos y se escoge la mejor.

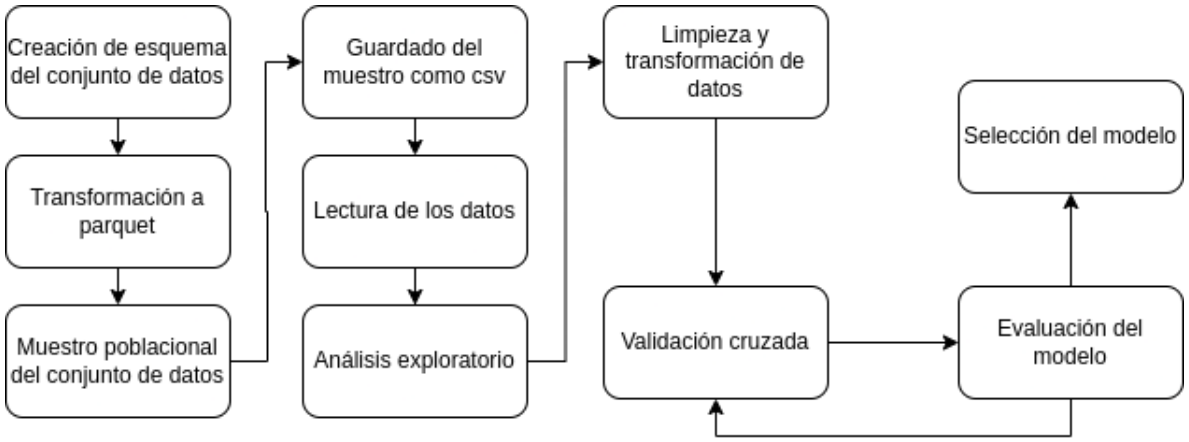

Figura 7. Diagrama de flujo del pipeline del proceso.

4.2 PREPROCESAMIENTO

El preprocesamiento de los datos, se realizó siguiendo las recomendaciones de Vahid Mirjalili y Sebastián Raschka [3]. Uno de los mayores retos de este conjunto de datos y la razón principal por la cual los resultados de la competencia no fueron favorables (incluso los competidores del top 3 no lograron llegar al 5% del MAPE), son los mismos datos. Los datos son la mayor dificultad de este reto, por ejemplo al analizar el gráfico de correlación de una de las iteraciones (figura 8), podemos notar que ninguna variable del conjunto de datos tiene una correlación mayor al 0.3, lo cual nos da un indicio de lo desafiante que será el diseño de un modelo con buen rendimiento.

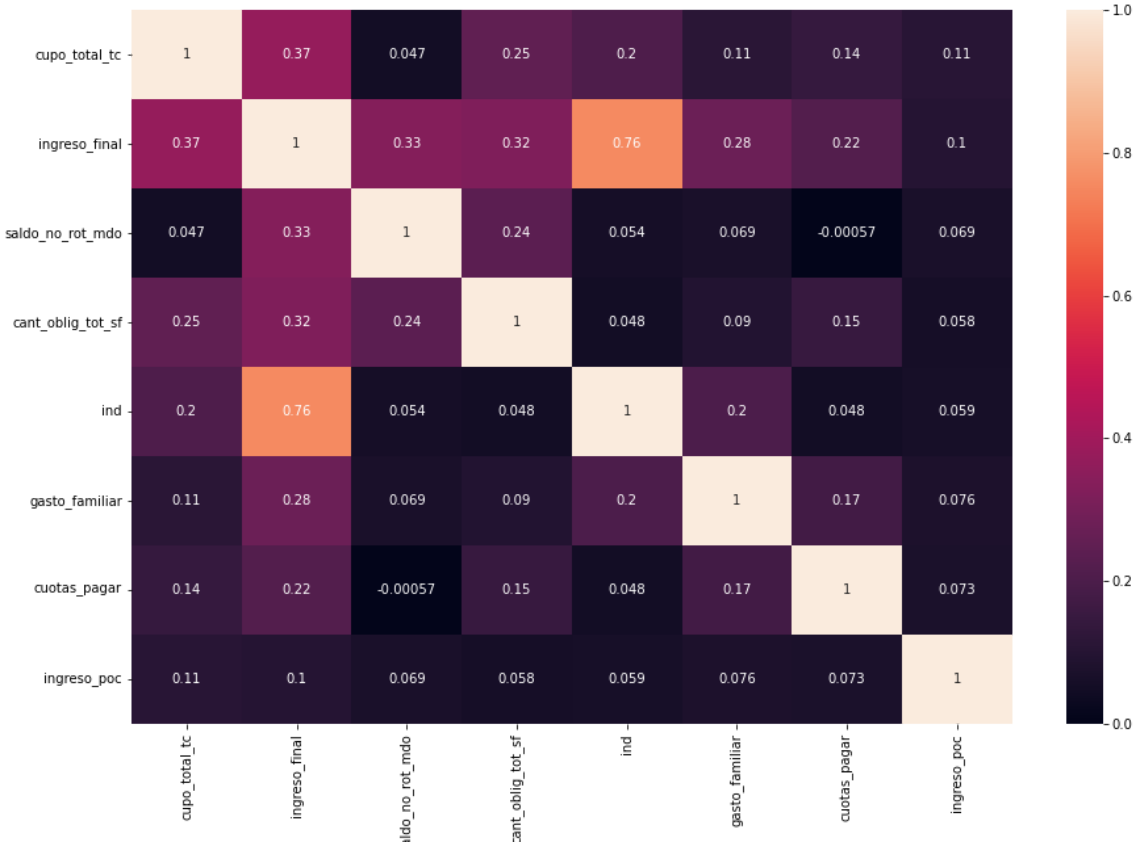

Figura 8. Matriz de correlación para los datos numéricos.

Primero se utilizó Apache Spark para la lectura de todos los datos de manera local, se creó el esquema de tipos y se convirtieron los datos a formato parquet para poder guardar muestras del conjunto de datos de una manera más rápida. Más adelante, se realizó un procedimiento de eliminación de categorías nulas y limpieza del dataset, en esta parte la conversión de los datos a parquet tomó un sentido más relevante.

Utilizando la muestra de datos que se extrajo en el paso anterior, se procedió a mirar la distribución de cada una de las variables utilizando histogramas y gráficos de barra, para así, poder determinar si existía un sesgo en los datos o determinar la cantidad de valores nulos que existían en cada variable. Con los gráficos fue posible determinar la limpieza necesaria y las transformaciones sobre las variables (Binarización o dummies).

En cuanto a las variables numéricas hay varias observaciones a realizar. La primera, existían variables con una gran cantidad de valores nulos, pero, al observar la naturaleza de las variables se consideró simplemente imputarlas con un 0 (cuota de crédito de consumo, cuota de crediagil, cuota de vivienda, etc). Segundo, a pesar de no presentar valores faltantes, existían muchas variables numéricas con valores nulos que realmente deben ser nulas (clientes que no habían tenido moras en algún momento o clientes sin créditos con el banco), el problema con dichas variables es que presentaban un desbalance similar a las variables categóricas, ese desbalance se traduce en variables con menor capacidad de descripción de la variable objetivo. Tercero, en las variables como el gasto familiar, el ingreso neto disponible y el valor de la cuota por crédito de consumo, existían valores negativos, los cuales no representan valores lógicos en el conjunto de datos, dichos valores se cambiaron por nulos. Por último, una gran cantidad de valores nulos y outliers en la variable objetivo, este inconveniente es el más importante a considerar, dado que es la razón principal por la cual la métrica mape llega a ser tan sensible, existen desde valores muy altos, hasta clientes con gastos en 0. Para solucionar dicho inconveniente tanto con los outliers de la variable objetivo como con las demás mencionadas, se eliminaron estos registros utilizando percentiles al 95% y al 5%.

Por otro lado, se realizaron diferentes simplificaciones sobre los datos. Sobre las variables categóricas se realizó binarización de algunas variables que solo implican una elección de sí o no. Con esta binarización se utilizaron todos los checks que implican la existencia de algún crédito con el banco, con esto, se creó el flag “crédito activo” y se eliminaron las demás variables. De igual manera, se realizaron simplificaciones de variables categóricas con múltiples categorías, simplificandolas en un solo flag del tipo si o no, por ejemplo, la variable nivel_academico se sustituyó con la variable ingreso_profesional, dicha variable indica si una persona recibe sus ingresos por conceptos profesionales (es decir, si es universitario o si tiene alguna especialización) o no, este mismo criterio se aplicó a las siguientes variables; estado_civil por comparte_gastos (es decir, si tiene pareja puede que compartan gastos) y ocupación por contrato_def_indef (esto es, si es empleado o no). En cuanto a las variables numéricas, se simplificaron las variables que indican el valor de la cuota que el cliente debía pagar por algún concepto crediticio y se generó la variable “cuotas por pagar”, en una forma más matemática, la variable cuotas por pagar se encuentra definida por la suma de las siguientes variables: cuota_tc_bancolombia, cuota_cred_hipot, cuota_de_consumo, cuota_rotativos,

cuota_tarjeta_de_credito, cuota_de_sector_solidario, cuota_libranza_sf y cuota_tc_mdo. Además, se realizó la simplificación de las variables que representaban algún tipo de ingreso, variables como ingreso por nómina y pensión, ingreso por pensión, ingreso social e ingreso final, realmente se encontraban relacionadas, sumando variables y chequeando los valores máximos entre ellas, se creó la variable "ingreso corregido" la cual permitió reducir cuatro variables en una sola.

Finalmente, realizando un análisis de correlación, se eliminaron todas las variables que presentaban una correlación mayor a 0.6, esto con el fin de analizar si había cambios en el rendimiento del modelo, estas y otras variables eliminadas se encuentran en la tabla 4. En este análisis se encontró que había correlación en variables que mencionan las moras de los clientes, así que muchas fueron eliminadas. Con todas estas simplificaciones quedó un conjunto de datos de aproximadamente 40 características de las 65 iniciales (eso sin contar como aumentan después al realizar One Hot Encoding), disminuyendo en gran medida el conjunto de datos original.

Variables eliminadas	
Nombre	Motivo
departamento_laboral	Demasiadas categorías para utilizar con un one hot encoder
departamento_residencia	Demasiadas categorías para utilizar con un one hot encoder
ciudad_laboral	Demasiadas categorías para utilizar con un one hot encoder
ciudad_residencia	Demasiadas categorías para utilizar con un one hot encoder
profesion	Demasiadas categorías para utilizar con un one hot encoder
cartera_castigada	No aporta suficiente información, a la mayoría de los clientes no se les ha castigado la cartera
rechazo_credito	No aporta suficiente información, a la mayoría de los clientes no se les ha rechazado algún tipo de crédito
tiene_ctas_embargadas	No aporta suficiente información, casi no hay clientes con cuentas embargadas
pension_fopep	Se encuentra de manera implícita con su par numérico
cat_ingreso	Se eliminó por presentar demasiados valores nulos
tipo_vivienda	Se eliminó por presentar demasiados valores nulos
rep_calif_cred	Su definición no aporta para la variable objetivo

estado_civil	se transformó en la variable comparte gastos
nivel_academico	se transformó por la variable ingresos_profesionales
ocupacion	se transformó por la variable contrato_indef_def
convenio_lib	No aporta información para la variable objetivo
tiene_consumo	Se transformó en la variable credito_activo
tiene_crediagil	Se transformó en la variable credito_activo
tiene_cred_hipo_1	Se transformó en la variable credito_activo
tiene_cred_hipo_2	Se transformó en la variable credito_activo
cuota_tc_bancolombia	Se eliminó para crear la variable total_cuotas
cuota_cred_hipot	Se eliminó para crear la variable total_cuotas
cuota_de_consumo	Se eliminó para crear la variable total_cuotas
cuota_rotativos	Se eliminó para crear la variable total_cuotas
cuota_tarjeta_de_credito	Se eliminó para crear la variable total_cuotas
cuota_de_sector_solidario	Se eliminó para crear la variable total_cuotas
cuota_libranza_sf	Se eliminó para crear la variable total_cuotas
cuota_tc_mdo	Se eliminó para crear la variable total_cuotas
cuota_de_vivienda	Se eliminó para crear la variable total_cuotas
cuota_sector_real_comercio	Se eliminó para crear la variable total_cuotas
ingreso_segurida_social	Se utilizó para la variable ingreso_corregido
ingreso_oc	Variable utilizada para crear la variable ingreso_corregido
ingreso_nomina	Se utilizó para la variable ingreso_corregido
ingreso_nompen	Se utilizó para la variable ingreso_corregido
ingreso_final	Se utilizó para la variable ingreso_corregido
id_cli	No aporta información para la variable objetivo
fecha_nacimiento	No aporta información para la variable objetivo
ult_actual	No aporta información para la variable objetivo

codigo_ciiu	No aporta información para la variable objetivo
cant_moras_30_ult_12_meses	Se eliminó porque presenta alta correlación con las otras variables a 60 y 90 días
cant_moras_60_ult_12_meses	Se eliminó porque presenta alta correlación con las otras variables a 30 y 90 días
cant_moras_90_ult_12_meses	Se eliminó porque al final conceptualmente hablando, esta variable no influye en la variable objetivo
ingreso_corregido	Se eliminó por presentar alta correlación con la variable ind. Se decidió quedarse con la ind por ser una variable recomendada por el personal del banco en la competencia.
tiene_ctas_activas	Se eliminó porque tiene alta correlación con la variable ctas_activas y nro_tot_cuentas
nro_tot_cuentas	Se eliminó porque tiene alta correlación con la variable ctas_activas y mediana_pem3
periodo	No aporta información para la variable objetivo

Tabla 4. Variables eliminadas y su motivo.

Por último, en la tabla 5 se presentan las variables finales para la última iteración.

Variables utilizadas	
edad	cupo_tc_mdo
genero	saldo_prom3_tdc_mdo
categoria	saldo_no_rot_mdo
ind_mora_vigente	cant_oblig_tot_sf
mora_max	cant_cast_ult_12m_sr
cupo_total_tc	ind
tenencia_tc	pol_centr_ext
ctas_activas	gasto_familiar
ctas_embargadas	comparte_gastos
mediana_nom3	ingreso_profesional

mediana_pen3	contrato_def_indef
cant_mora_30_tdc_ult_3m_sf	credito_activo
cant_mora_30_consum_ult_3m_sf	total_cuotas

Tabla 5. Variables finales para el entrenamiento de los modelos durante la última iteración.

4.3 MODELOS

Se utilizaron un total de cuatro modelos, entre ellos dos de ensambles, uno lineal y uno de máquina de soportes vectoriales, sus configuraciones fueron de la siguiente manera:

- **ElasticNet:** Modelo lineal que permite añadir parámetros de regularización L1 y L2 [4], este modelo fue entrenado con los parámetros más óptimos estimados por la propia librería de scikit learn.
- **SVR:** Modelo basado en máquinas de soporte vectorial especializado en problemas de regresión. Este modelo también se entrenó utilizando los parámetros más óptimos estimados por la librería de scikit learn, el parámetro de regularización C se utilizó como 1, y el kernel fue un rbf, dada la complejidad de los datos utilizados, se utilizó el kernel que mejor resultados ofrece para conjuntos de datos más complejos.
- **Random Forest Regressor:** Modelo de ensamble basado en árboles de decisión con técnica bagging [5], este modelo se utilizó de muchas maneras, variando principalmente su número de estimadores y la profundidad máxima de cada árbol. Se utilizaron estimadores con 50, 150, 300 y 500 árboles de decisión y sus profundidades se complejizan desde 6, 15, 30 y 60 en su nivel de profundidad.
- **Gradient Boost Tree:** Modelo de ensamble basado en árboles de decisión con técnica boosting [6], la complejidad del modelo fue variando de la misma manera que lo hizo el modelo de Random Forest.

4.4 MÉTRICAS

Las métricas para medir el rendimiento de los modelos se calcularon de la siguiente manera; para el caso de los modelos divididos utilizando train test split, se utilizaron las funciones r2_score y mean_average_percentage_error del módulo metrics de la librería de scikit-learn, y, para la validación cruzada, se implementaron las mismas métricas utilizando la función cross_validate del módulo de model selection, especificando sus nombres en el campo scoring.

5. METODOLOGÍA

5.1 BASELINE

Para la primera iteración, el primer inconveniente que se encontró fue el formato de los datos, los cuales se encontraban en archivos csv separados por comas. El inconveniente con ese formato estaba relacionado con la columna Profesión, ya que presentaba categorías separadas por comas, es decir, el mismo separador que el de las columnas en el conjunto de datos. Esto indicaba que al momento de realizar la carga de datos se iban a encontrar filas con una mayor cantidad de columnas. Para solventar este inconveniente se utilizó la herramienta excel y se encontraron los patrones que incluyen este separador con el fin de ser cambiado. De lo anterior, excel generó un conjunto de datos recortado del total del dataset y por simplicidad, este fue el utilizado para la primera prueba.

Con la muestra de datos se realizaron diferentes trabajos de limpieza de datos los cuales incluyeron: imputación de valores nulos en variables categóricas (categoría sin información ó no informa), eliminación de datos nulos que no podían ser imputados, eliminación de columnas que no brindaban información relevante al modelo (id del cliente, código ciiu, periodo, etc), conversión de tipos, imputación de valores nulos en variables numéricas, filtración de edades entre los 20 y 70 años de edad, y reducción de variables numéricas sumando todas las cuotas y simplificándolas en una sola característica llamada cuotas a pagar. Se decidió trabajar el primer modelo utilizando un algoritmo de random forest (200 estimadores y profundidad del árbol en 30), teniendo en consideración sólo las variables numéricas. Los resultados de este modelo fueron del orden 2x10^20, es decir, un error grande debido a los outliers y valores negativos presentes en la variable objetivo, al momento de calcular el error, estos outliers generaron grandes valores residuales al comparar las predicciones con los datos reales.

El segundo modelo que se probó fue un modelo utilizando solamente las variables categóricas, cuyos resultados fueron similares al modelo que solo incluía variables numéricas. Nuevamente los outliers de la variable target generaron ese impacto en la métrica de medición. Estos inconvenientes fueron solucionados en las siguientes iteraciones donde se trabajó mucho más fuerte la componente de los datos y el feature engineering.

5.2 VALIDACIÓN

El proceso de validación de los modelos se trabajó de dos maneras independientes. La primera, diseñando los modelos a evaluar, realizando una partición de los datos entre 70/30 para los conjuntos de entrenamiento y prueba respectivamente. Cada uno de los modelos que se probó

tenía la misma configuración y utilizando la misma semilla para que la partición fuera reproducible y se tuviera un punto de comparación.

El otro proceso de validación, se realizó cargando los datos en la función cross_validate de scikit learn, la cual permitió trabajar con validación cruzada para tener un mejor conocimiento del rendimiento del modelo a lo largo de todo conjunto de datos. Además, este modelo se utilizó para realizar predicciones, crear las gráficas de predicción y valores residuales, la cual nos permitió comprender de manera gráfica qué estaba ocurriendo en el modelo a nivel de las estimaciones que realiza.

La primera prueba de validación permitió confirmar que los modelos eran sensibles a la partición de los datos (gracias al apoyo del coeficiente de determinación R2), pues los valores del MAPE en algunas particiones llegaba a ser 5 o 60 o incluso 200, por ello, se decidió descartar la prueba por división de datos en entrenamiento y prueba y utilizar la validación cruzada junto con las gráficas de valores residuales, esto permitió descartar cualquier valor erróneo de "Buen desempeño".

5.3 ITERACIONES Y EVOLUCIÓN

Para las siguientes iteraciones se inició con un análisis exploratorio más profundo y detallado para analizar qué estaba ocurriendo a nivel de los datos. Los resultados permitieron comprender que el problema no se encontraba realmente en la variable objetivo, como se pensaba en la primera iteración. Los resultados demostraron que los datos de la competencia se encontraban completamente sesgados a una población en específico, las categorías se encontraban desbalanceadas, las variables numéricas tenían muchos valores atípicos y otros erróneos, algunas variables presentaban sólo un 10% de datos numéricos y el resto eran valores nulos. Todos los inconvenientes mencionados, requieren que las iteraciones se centran más sobre los datos que en los propios procesos de machine learning. Entre las iteraciones más destacadas se encuentran:

- Se realizó la limpieza del conjunto de datos, se binarizaron variables categóricas simples y se obtuvieron las dummies de las variables más complejas, sin tener en cuenta variables como la ciudad de residencia o el departamento laboral, que incluyen demasiadas categorías. Se realizó análisis de outliers sobre la variable target y se simplificaron algunas variables numéricas como las cuotas de los créditos en una sola variable llamada cuotas a pagar. Además, se tomaron los valores negativos de la variable cuota de consumo con su contraparte positiva. Finalmente, se evaluaron modelos como Gradient Boost Tree, ElasticNet y SVR.

- Se tomó todo el conjunto de datos por completo y se eliminaron de diferentes variables categóricas: las categorías nulas o que no brindaban información (“no informa”, “sin información”) para crear un conjunto de datos llamado “Dataset de fuerza bruta”, ya que obligaba a las variables categóricas a ajustarse a las condiciones más ideales posibles. De igual manera, se obviaron las variables que indican algún tipo de mora, se categorizó la edad, se trabajó con las sumas de los ingresos y la suma de las cuotas, con este dataset se realizaron pruebas en modalidad cross validate con los modelos ElasticNet y Gradiente Boost Tree. Este dataset llamado fuerza bruta, se generó utilizando Spark.
- Se trabajó desde cero con el conjunto de datos utilizando una muestra aleatoria representativa al conjunto de datos original, algunas variables categóricas se transformaron en variables binarizadas como; estado civil se transformó a comparte_gastos, nivel académico se transformó a ingreso_profesional, ocupación se transformó en contrato_def_indef y las variables que verifican la tenencia de un crédito o no, se transformaron en una sola variable que verifica si el cliente tiene algún crédito activo con el banco. En cuanto a las variables numéricas, se eliminaron valores negativos de las variables crédito de consumo y ingreso neto disponible, se eliminaron de outliers en las variables de ingreso neto disponible y gasto familiar, se realizó correcciones a la variable ingreso final, utilizando las otras variables de ingreso (nómina, nómina y pensión, y ingreso social),se simplificaron las variables asociadas con las cuotas en una sola variable llamada cuotas a pagar. Se incluyó el modelo de random forest para analizar junto con los demás.
- En esta última iteración, se trabajó con los datos del departamento de Antioquia para analizar el performance de los modelos centrado en este conjunto que cuenta con aproximadamente el 25% de los datos de toda la muestra.

5.4 HERRAMIENTAS

Las herramientas utilizadas para este proyecto y sus usos fueron las siguientes:

- **Python ->** Lenguaje de programación más popular para trabajos de ciencia de datos.
- **Spark(PySpark) ->** Framework de Big Data para manejo de grandes volúmenes de información ya sea en un cluster de servidores o en una máquina local.
- **Pandas ->** Librería de python para manejo de datos tabulares
- **Numpy ->** Librería de python optimizada para cálculos matemáticos
- **Scikit-learn ->** Librería de Python para trabajar con modelos de machine learning
- **Matplotlib ->** Librería de Python para trabajar con visualización de datos
- **Seaborn ->** Librería de Python para trabajar con visualización de datos de una manera visual más agradable.
- **Vertex AI ->** Máquinas virtuales de google para crear instancias con Jupyter Lab.

6. RESULTADOS

6.1 MÉTRICAS

Segunda Iteración

El resultado del performance durante esta segunda iteración se encuentra en las tablas 6 y 7. La tabla 6 es un punto de comparación entre el método train/test y el cross validate, donde resultados como el performance del modelo gradient boost tree sencillo, permite concluir que la metodología de train/test no funciona muy bien en este caso debido a que los modelos son demasiado sensibles a la partición de los datos. Por otro lado, el mejor rendimiento alcanzado fue el del modelo gradient Boost Tree sencillo, al centrarnos solo en el coeficiente R2, podemos apreciar que es el modelo cuyas predicciones se encuentran más cercanas a la línea de ajuste del modelo.

Modelo	Performance train/test		Performance cross validate	
	R2	MAPE	R2	MAPE
ElasticNet	0.0481	101.4325	0.0497	107.5608
GradientBoostTree Sencillo	0.1228	6.5469	0.1244	109.0989
GradientBoostTree Intermedio	0.0411	87.5456	0.0396	109.1727
GradientBoostTree complejo	-0.0168	70.7241	--	--

Tabla 6. Métricas de evaluación para la segunda iteración, método train test split y cross validate.

Analizando las métricas evaluadas, se determinó no usar metodología train/test y se procedió solo con el uso del cross validate. La tabla 7 muestra algo interesante, el rendimiento del modelo sobre el propio conjunto de datos de entrenamiento puede aumentar al complejizar el modelo, pero el rendimiento en los datos de prueba solo disminuye. Este resultado permite aceptar y concluir con la hipótesis inicial de este trabajo, los datos de la competencia por sí solos no funcionan para el objetivo del problema.

Modelo	Performance entrenamiento		Performance prueba	
	R2	MAPE	R2	MAPE
ElasticNet	0.0506	107.4420	0.0497	107.5608
GradientBoostTree Sencillo	0.1381	111.7188	0.1244	109.0989
GradientBoostTree Intermedio	0.7051	46.4300	0.0396	109.1727
GradientBoostTree complejo	-	-	-	-

Tabla 7. Métricas de evaluación para la segunda iteración, método cross validate con métricas sobre el conjunto de entrenamiento.

Para comprender qué está sucediendo con los datos, se realizaron gráficas de comparación entre las predicciones y los datos reales. La figura 9 muestra la comparación de las predicciones realizadas con el mejor modelo (Gradient Boost Tree sencillo con 50 estimadores y 6 de profundidad máxima) en forma de un histograma, hay un desfase muy pronunciado y podemos ver como las predicciones solo se realizan sobre una pequeña fracción del grupo de datos original, esto indica que el modelo no está aprendiendo hacia los extremos, ni valores muy bajos ni valores muy altos.

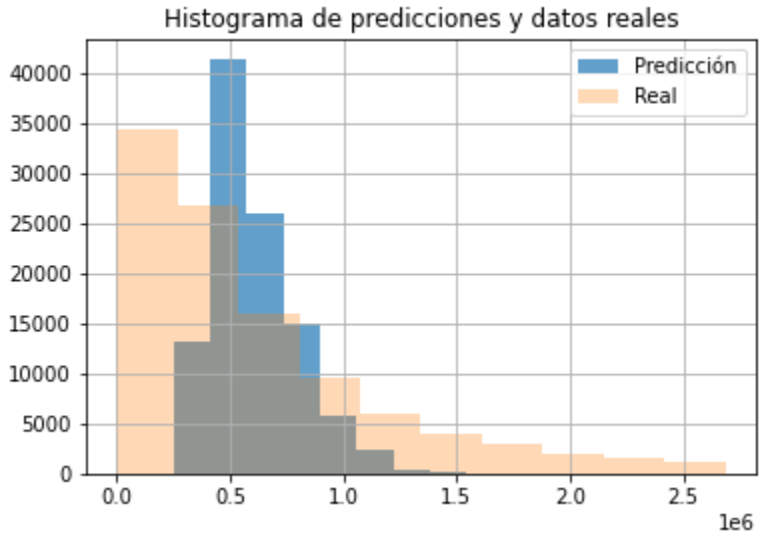

Figura 9. Histograma de las predicciones y los datos reales, segunda iteración.

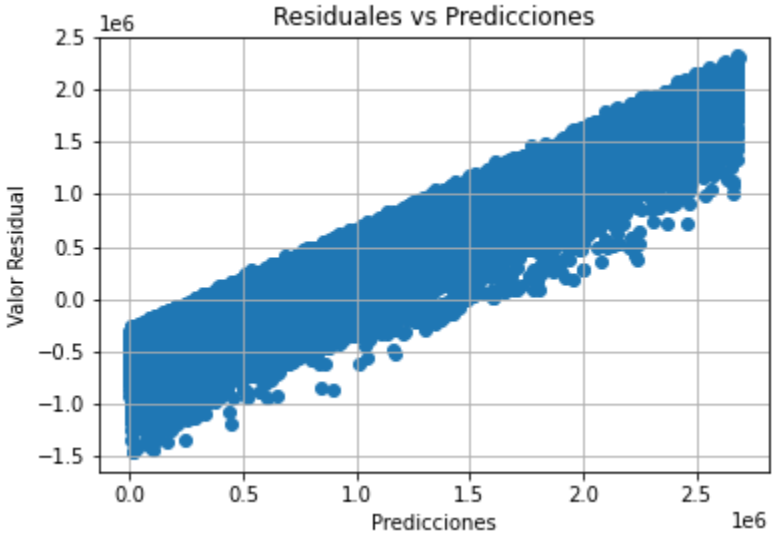

Figura 10. Gráfica de los residuales y las predicciones, segunda iteración.

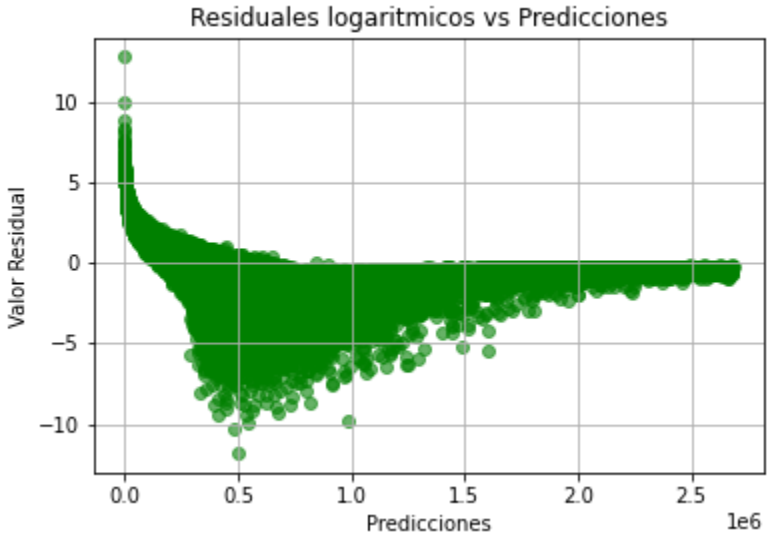

Figura 11. Gráfica de los residuales logarítmicos y las predicciones, segunda iteración.

La figura 10 y la figura 11, nos permiten observar a un nivel más detallado qué ocurre con las predicciones. Las gráficas de residuales presentadas nos permiten observar que los residuales de los datos tienen un patrón lineal, la gráfica que se esperaría ver en la figura 10, sería una agrupación de datos en el nivel 0 del valor residual, pero por el contrario se encuentra que los residuales van aumentando de manera lineal. Además, en la figura 11 se puede observar que los

valores que más afectan al modelo, son los valores de los gastos familiares más bajos, esto se debe a que existen clientes que presentan un gasto familiar mínimo cercano a 0.

Cuarta Iteración

Durante la tercera iteración no se encontraron resultados significativos, el trabajo realizado sobre el conjunto de datos no dio resultados diferentes a los que se tienen con la segunda iteración, ello permite concluir que la modificación de los datos eliminando categorías nulas, no es un buen enfoque para resolver el problema de los datos. Así mismo, durante la cuarta iteración, se trabajó con muchas modificaciones sobre los datos entre las cuales se encuentran; binarización de variables categóricas, imputación de valores nulos (variables categóricas y numéricas), reducción de variables binarias que indican deudas crediticias (tiene crédito de consumo, tiene crediagil, tiene crédito hipotecario, etc), corrección del ingreso final (basado en el ingreso por nómina, ingreso por nómina y pensión, etc), reducción de variables categóricas con múltiples categorías en una sola categoría (ocupación, estado civil y nivel académico), eliminación de outliers (mediante los percentiles 95 y 5) y reducción de variables con alta correlación con otras variables (entre ellas, cupo total en tarjeta de crédito, cantidad de obligaciones financieras, si tiene cuentas activas, etc), pero no se logró algún cambio importante. Los datos siguen mostrando el mismo comportamiento, entre más complejo es el modelo, tiende a perder su capacidad para ajustarse a los datos.

Modelo	Performance entrenamiento		Performance prueba	
	R2	MAPE	R2	MAPE
ElasticNet	0.0670	88.3518	0.0665	88.2130
GradientBoostTree Sencillo	0.1395	91.4026	0.0935	93.2056
GradientBoostTree Intermedio	0.5541	43.8189	0.0328	60.9157
GradientBoostTree complejo	0.9861	7.1962	-0.0700	59.0081

Tabla 8. Métricas de evaluación para la cuarta iteración con método cross validate.

Quinta Iteración

En esta última iteración no se obtuvieron resultados prometedores con las modificaciones realizadas sobre los datos. Un resultado que permite confirmar la hipótesis del sesgo poblacional

de los datos, ocurre cuando nos centramos en realizar un modelo solamente con los datos de la población de Antioquia. Analizando la tabla 9, no se puede observar algo en particular, solamente modelos que han sido sobre ajustados a los datos de entrenamientos con el fin de obtener al menos un mejor score en los datos de prueba. Por otro lado, al analizar la figura 12, vemos que el desfase de los histogramas ha disminuido (utilizando el gradient boost tree con 500 estimadores y profundidad máxima de 15), esto afirma la hipótesis y es necesario agregar más datos que permitan disminuir este sesgo.

Modelo	Performance entrenamiento		Performance prueba	
	R2	MAPE	R2	MAPE
ElasticNet	0.0031	77.0626	0.0030	77.1377
GradientBoostTree Sencillo	0.5541	43.8189	0.0328	60.9157
GradientBoostTree intermedio	0.8273	80.3441	0.0565	-190.1892
GradientBoostTree complejo	1.0000	0.0000	-0.0779	110.1827
RandomForest intermedio	0.8274	80.3441	0.0565	190.1893

Tabla 9. Métricas de evaluación para la quinta iteración con método cross validate.

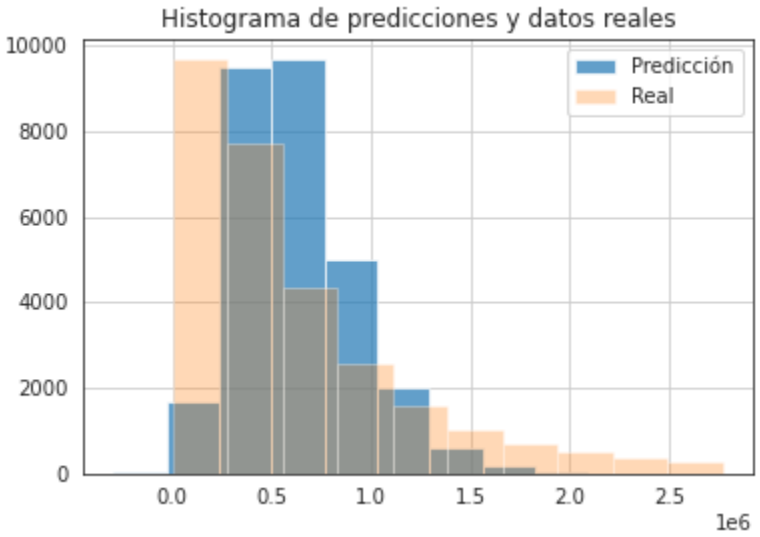

Figura 12. Histograma de las predicciones y los datos reales, quinta iteración.

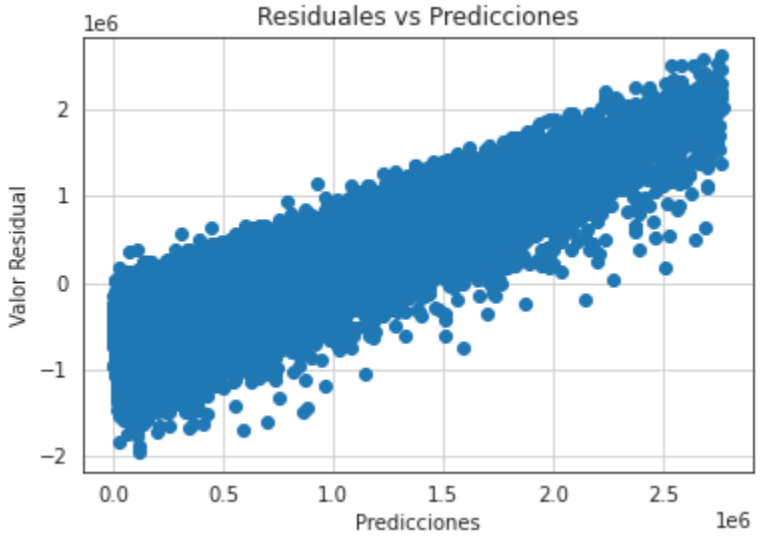

Figura 13. Gráfica de los residuales y las predicciones, quinta iteración.

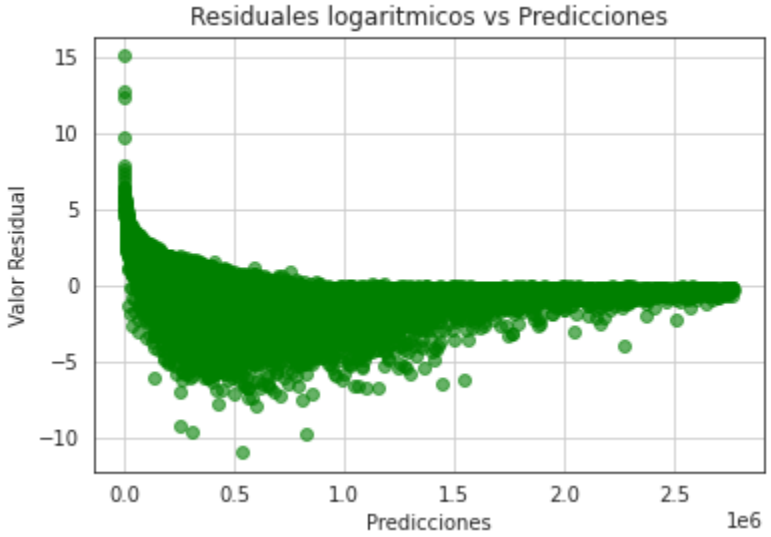

Figura 14. Gráfica de los residuales logarítmicos y las predicciones, quinta iteración.

6.2 EVALUACIÓN CUALITATIVA

Los resultados obtenidos permiten concluir que se debe de trabajar con otro tipo de datos, ya que los modelos implementados no son capaces de ajustarse. Cuando se propone la implementación de un modelo sencillo, las métricas de desempeño presentan valores muy bajos, esto se puede apreciar en las tablas de la 6 a la 9 donde el modelo ELasticNet o el Gradiente Boost Tree con menos complejidad, presentan valores que no superan el 20% del R2 (el mape es demasiado sensible para los datos y muestra resultados muy variantes). Los resultados más interesantes pueden observarse en las figuras 11 y 14, donde los valores de los residuales de cada predicción

muestran de una forma fácil, la enorme diferencia que se generan entre las predicciones y los valores reales.

Por otro lado, al implementar modelos más complejos (ver tablas de la 6 a la 9) se nota un incremento en las métricas de desempeño para los conjuntos de datos de entrenamiento, pero el desempeño de las pruebas continúa disminuyendo. Se realizaron transformaciones a los datos, lo cual indica que el problema no radica en los modelos, sino, en los datos.

Los resultados obtenidos por las diferentes configuraciones de los datos y de los modelos, no permiten concluir con la puesta en producción de un modelo para este problema, lo cual dificulta la evaluación de la métrica de negocio propuesta.

6.3 CONSIDERACIONES DE PRODUCCIÓN

Como se mencionó anteriormente, no existe un antecedente para el cálculo de la métrica de negocio. Pero se podría calcular de la siguiente manera; el modelo sería desplegado en un servicio en la nube como cloud run de GCP o api gateway de AWS, cada vez que se ingrese información nueva de los clientes el modelo realizará la predicción de sus gastos familiares. Esta información será presentada al analista del banco quien definirá si proponer un crédito, una tarjeta de crédito o simplemente ofrecerá algún producto del banco a dicho cliente. Si el cliente accede a tomar el producto entrara en una base de datos de nuevos productos crediticios por incidencia del modelo generado, y por último, al final del mes, se realizará un conteo total de los clientes que han adquirido nuevos productos crediticios, el dinero generado y un porcentaje de ajuste a su realidad financiera. De esta manera, se puede evaluar qué tantos ingresos está generando el banco con la obtención de nuevos productos gracias al modelo generado.

Para una posible puesta en producción se recomienda el uso integrado de google cloud run, google analytics y google data studio. El modelo se guardará utilizando la librería joblib, se creará una api utilizando ya sea Flask o FastAPI, que permitirá consumir el modelo como un servicio, esta api se pondrá en un contenedor de docker para su despliegue en cloud run.

Cada vez que haya una actualización de los datos de los clientes, los datos serán enviados al modelo para realizar predicciones, esas predicciones se convertirán en aliados para que los analistas puedan determinar que producto crediticio ofrecer a los clientes (campaña comercial). Los resultados de las acciones tomadas, se analizaron con Google analytics y se crearán dashboards que incluyan la métrica de precisión del modelo en google cloud studio. Todos los datos que se envíen a predecir, luego serán validados y con esa validación se corroboró el estado del modelo, para analizar periódicamente los reentrenamientos que deba tener.

7. CONCLUSIONES

- Los resultados de performance de los modelos tanto para modelos sencillos como modelos complejos, y junto con los resultados del análisis exploratorio de los datos, indican que el conjunto de datos utilizados para este trabajo no son los mejores dado que se encuentran sesgados y con variables desbalanceadas. Los modelos demostraron que a pesar de estar sobre ajustados, la métrica de evaluación no mejoraba para el conjunto de prueba, solo empeoraba, esto indica que los modelos no eran capaces de ajustarse al conjunto de datos del problema.
- La competencia propuso el MAPE como métrica de evaluación, pero al realizar los primeros experimentos con la metodología de particionar los datos en conjuntos de entrenamiento y prueba, se encontró que esta métrica era sensible al conjunto de datos con los que se realizaban las pruebas. Además, podría entregar resultados de alta precisión que realmente no se ajustan al conjunto de datos. Por lo anterior, todas las pruebas se realizaron en conjunto con el coeficiente de determinación (R2), el cual demostró dar un valor más acertado del rendimiento del modelo.
- Los datos propuestos para este reto tendrían una mayor significancia para problemas de otro tipo, como clasificar el rechazo o aceptación de un producto crediticio. Lo anterior debido a que se cuenta con suficiente información financiera que permite describir en gran medida ese tipo de problema. Las moras, carteras castigadas, cuotas de créditos, todas son variables que funcionan para un problema de clasificación.
- Como recomendaciones correctivas al dataset se tienen: mejorar el balance de las categorías en las variables de tipo categórico (para así evitar un sesgo poblacional), datos completamente actualizados, cuidar la calidad de los datos al momento de llenar la información de un nuevo cliente, evitando las categorías “no informa” o “sin información”. A futuro podrían incluirse en el dataset, ciertos índices económicos que permitan ampliar las características que describen mejor a la variable objetivo, variables como el icc o el precio del dólar, pueden generar mayor información sobre la realidad económica del país y por ende, dar mayor aporte a variables como el departamento de residencia y a la variable objetivo, por ejemplo.
- El modelo más óptimo de este proyecto, no logra llegar a los resultados obtenidos por los participantes de la competencia, en el top 3 llegan a un MAPE de aproximadamente el 57%, mientras el MAPE del proyecto llega a un 109% (verificado junto con la métrica R2). En el proyecto solo se trabajaron los datos suministrados por la competencia, mientras los competidores pueden haber incluido varias fuentes de datos.

8. BIBLIOGRAFÍA

[1]. Cameron, A. C., & Windmeijer, F. A. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of econometrics*, *77*(2), 329-342.
[2]. Khair, U., Fahmi, H., Al Hakim, S., & Rahim, R. (2017, December). Forecasting error calculation with mean absolute deviation and mean absolute percentage error. In *Journal of Physics: Conference Series* (Vol. 930, No. 1, p. 012002). IOP Publishing.
[3]. Mirjalili, V., & Raschka, S. (2020). *Python machine learning*. Marcombo.
[4]. Hans, C. (2011). Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, *106*(496), 1383-1393.
[5]. Segal, M. R. (2004). Machine learning benchmarks and random forest regression.
[6]. Prettenhofer, P., & Louppe, G. (2014). Gradient boosted regression trees in scikit-learn.