



UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DIRECCIÓN DE POSGRADO



**DIPLOMADO ESTADÍSTICA APLICADA A LA TOMA
DE DECISIONES**
SEGUNDA VERSIÓN

PRÁCTICA #2

NOMBRE : HEBERT JUAN DE DIOS DELGADILLO FERNANDEZ
CARLOS ALFREDO ORIHUELA BERRIOS
DOCENTE : DANNY LUIS HUANCA SEVILLA

Cochabamba – Bolivia

Contenido

1	ANTECEDENTES.....	4
2	IDENTIFICACION DEL PROBLEMA	4
3	FASE DE ENTENDIMIENTO DEL NEGOCIO.....	5
3.1	Desde una perspectiva comercial.....	5
3.1.1	¿Qué espera obtener de este proyecto?.....	5
3.1.2	¿Cómo define la finalización de los trabajos?	5
3.1.3	¿Dispone de la dotación presupuestaria y de los recursos necesarios para completar los objetivos?	5
3.1.4	¿Dispone de acceso a todos los datos necesarios para el proyecto?.....	5
3.1.5	¿Ha tratado con su equipo los riesgos y contingencias asociadas con el proyecto?	6
3.1.6	¿Los resultados del análisis de coste/beneficios hacen que el proyecto sea viable?	6
3.2	Desde una perspectiva de ciencia de datos	6
3.2.1	¿En qué forma puede ayudarle la ciencia de datos a cumplir sus objetivos comerciales?.....	6
3.2.2	¿Sabe qué técnicas de ciencia de datos producen los mejores resultados?.....	6
3.2.3	¿Cómo se implementarán los resultados de modelado? ¿Ha considerado implementar su plan de proyecto?.....	6
3.2.4	¿El plan de proyecto incluye todas las fases de CRISP-DM?	6
3.2.5	¿Los riesgos y dependencias se incluyen en el plan?	7
4	ENTENDIMIENTO DE LOS DATOS.....	8
4.1	Univariado con todas las variables que componen el dataset realizar lo siguiente:	8
4.2	Bivariado con las variables que ingresen en el estudio, respecto de la variable objetivo o target.	15
4.3	Generación un perfilado del total de variables que usará para la construcción del modelo.	22
4.3.1	¿Cuál es su nivel de comprensión de los datos?	23
4.3.2	¿Ha identificado y accedido correctamente a todos los orígenes de datos? ¿Ha tenido algún problema o restricción de algún tipo?	23
4.3.3	¿Ha identificado atributos clave de los datos disponibles?	23
4.3.4	¿Le han ayudado estos atributos a formular hipótesis?	24
4.3.5	¿Ha detectado el tamaño de todos los orígenes de datos?	24
4.3.6	¿Puede utilizar un subconjunto de datos cuando lo estime conveniente?.....	24

4.3.7	¿Ha calculado los estadísticos básicos de cada atributo de su interés? ¿Ha obtenido información de interés?	24
4.3.8	¿Ha utilizado gráficos de exploración para obtener atributos clave? ¿Este conocimiento ha reformulado alguna de sus hipótesis?	24
4.3.9	¿Cuáles fueron los problemas de calidad de datos del proyecto? ¿Tiene una planificación para resolver estos problemas?	24
4.3.10	¿Las fases de preparación de los datos son claras? Por ejemplo, ¿sabe qué orígenes de datos debe fusionar y los atributos que debe filtrar o seleccionar?	25
4.4	Subir los datos a DataBricks	¡Error! Marcador no definido.
5	Modelado	29
5.1	Aprendizaje Supervisado	30

1 ANTECEDENTES

El conjunto de datos de Berka, brinda información sobre los clientes, las cuentas y las transacciones del banco. La información dada es una colección de información financiera real en la cual se han anonimizado los datos sensibles por seguridad del banco. El banco almacena datos sobre sus clientes, las cuentas (transacciones de varios meses), los préstamos ya otorgados, las tarjetas de crédito emitidas.

2 IDENTIFICACION DEL PROBLEMA

El banco quiere mejorar sus servicios, mediante la identificación de quién es un buen cliente (a quién ofrecer algunos servicios adicionales) y quién es un mal cliente (a quién vigilar cuidadosamente para minimizar las pérdidas del banco). Los gerentes del banco esperan mejorar su comprensión de los clientes y buscan acciones específicas para mejorar los servicios.

INICIATIVAS DE NEGOCIO	
Implementa programas de fidelización que recompensen a los clientes leales y frecuentes. Esto puede incluir descuentos especiales, puntos acumulativos, reembolsos en efectivo u otros incentivos que alienten a los clientes a utilizar sus tarjetas de crédito del banco con mayor frecuencia.	
Implementar sistemas avanzados de detección de fraudes y monitoreo de transacciones para identificar y prevenir actividades fraudulentas. Esto puede ayudar a reducir las pérdidas asociadas con clientes malos o fraudes en tarjetas de crédito.	
Realizar análisis de riesgo crediticio más detallados utilizando datos financieros y no financieros de los clientes. Esto puede ayudar al banco a evaluar de manera más precisa la capacidad crediticia de los clientes y a evitar prestar a clientes con un historial crediticio deficiente.	
Ofrecer ofertas y promociones personalizadas a los clientes según sus necesidades y comportamiento de gasto. Para aumentar la satisfacción del cliente y fomentar un mayor uso de las tarjetas de crédito del banco.	
Brinda una experiencia del cliente excepcional en todos los puntos de contacto, ya sea en persona, en línea o a través de aplicaciones móviles. Esto incluye tiempos de espera reducidos, respuestas rápidas a consultas y problemas, y un servicio al cliente amigable.	
Entidades Clave y decisiones clave	
Entidad 1:	Entidad 2:
Cliente	Servicios del banco
¿Qué antigüedad tienen en promedio los clientes que si pagan sus deudas?	¿Cuál es la cantidad de transacciones hechas por mes de los clientes que pagan sus deudas?
¿Cuál es salario promedio de los clientes que pagan sus deuda con los que no?	¿Cuál es el porcentaje de tarjetas de credito otorgadas a los clientes que pagan su deuda?
¿Cuál es la cantidad de saldo promedio de los clientes?	¿Cuál es el prestamo mas comunmente solicitado por los clientes?
Cual es el presatamo promedio de los clientes?	¿Cuál es la tendencia de transacciones de los clientes que pagan sus deudas y de los que no?
Que distrito concentra la mayor cantidad de clientes del banco?	
Casos de Uso	Modelo Analítico Asociado
Identificar y clasificar a los clientes mas fiables para otorgar una tarjeta de credito.	Aprendizaje supervisado - Clasificación
Segmentar a los clientes por capacidad de pago.	Aprendizaje no supervisado - Clustering
Incluir descuentos especiales, puntos acumulativos, reembolsos en efectivo u otros incentivos que alienten a los clientes a utilizar sus tarjetas de crédito del banco con mayor frecuencia.	Aprendizaje no supervisado - Modelos de recomendación

3 FASE DE ENTENDIMIENTO DEL NEGOCIO

3.1 Desde una perspectiva comercial

3.1.1 ¿Qué espera obtener de este proyecto?

Lo que se espera obtener de este proyecto es que el banco quiere mejorar sus servicios de atención siendo capaz de identificar clientes potenciales, a los cuales ofrecer una línea de crédito con su respectiva tarjeta. Así como también reducir los riesgos de clientes que no terminan de pagar su deuda. Se espera clasificar a los tipos de cliente atractivos de los que no para actuar de forma distinta en ambos casos. Se pretende tener directrices en función de un análisis de datos para la clasificación de clientes aptos para el servicio de tarjeta de crédito.

3.1.2 ¿Cómo define la finalización de los trabajos?

Como un modelo terminado en el cual al registrarse la solicitud de un nuevo préstamo pueda alterarse la confiabilidad del cliente basado en el historial registrado del solicitante en el banco. La implementación, puesta en funcionamiento y supervisión del algoritmo en los servidores también forman parte de la finalización del trabajo.

3.1.3 ¿Dispone de la dotación presupuestaria y de los recursos necesarios para completar los objetivos?

Si, la descripción está realizada en la siguiente tabla, los costos de mantenimiento y supervisión no fueron considerados por solicitud del contratista, como ejemplo aproximado.

Presupuesto		
Equipo de trabajo	Experto en SQL y BBDD	Analista de datos
Salario total	BOB 197,500.00	BOB 181,700.00
Cuota patronal	BOB 63,397.50	BOB 58,325.70
Indemnización de finalización	BOB 3,950.00	BOB 3,318.00
Personas/mes	2	3
Duración laboral	140 horas	
Duración estimada	910 horas	
Costo del servidor SQL	BOB 6,883.44	
Configuración y conexión en red del servidor SQL	BOB 2,227.20	
Costos de transporte	BOB 835.20	
Gastos Varios	BOB 13,920.00	

3.1.4 ¿Dispone de acceso a todos los datos necesarios para el proyecto?

Si, el gerente en coordinación con los administradores de los servidores del banco nos dio acceso a todos los datos requeridos para el análisis y la ciencia de datos. El contacto con los mismos será mantenido en caso de requerir mayor información. Se tiene los datos necesarios para la realización de un modelo bajo las variables otorgadas, aun así, se puede precisar de mayor data con mayor rango de variables para un modelo de mayor confiabilidad y precisión de extrapolación. Por ejemplo, variables como Salario, Estado Civil, Ocupación, etc.

3.1.5 ¿Ha tratado con su equipo los riesgos y contingencias asociadas con el proyecto?

Si, los datos provistos serán respaldados cada determinado periodo de tiempo y las pruebas serán en bases de datos externas para poder realizar transformaciones y evitar percances ya que los datos de base no deben ser modificados bajo ninguna circunstancia, así mismo la transformación de los mismos debe ser confiable y no al azar ya que se tomarán decisiones en base a los mismos. En caso de existir modificaciones inesperadas o contratiempos, los datos serán recuperados de los respaldos.

3.1.6 ¿Los resultados del análisis de coste/beneficios hacen que el proyecto sea viable?

Resulta viable considerando el ahorro que se tendrá al evitar pérdidas por clientes morosos o que no paguen sus deudas a tiempo teniendo una relación estimada prevista de 0.85 de costo-beneficio es decir que se recuperara un 17.6% de la inversión por año considerando el costo de la implementación y de la compra de los equipos necesarios.

3.2 Desde una perspectiva de ciencia de datos

3.2.1 ¿En qué forma puede ayudarle la ciencia de datos a cumplir sus objetivos comerciales?

Se puede utilizar un algoritmo de Machine Learning de clasificación de tipo supervisado (reg. Logística, árboles de decisión, etc) gracias a que se cuenta con el historial de varios clientes y el manejo de su cuenta, así como también el estado de sus deudas anteriores. Todo esto para poder clasificar el tipo de cliente a quien otorgarle un crédito y tarjeta de crédito y a quien no o por lo menos otorgarle una línea de crédito baja con tarjeta junior.

3.2.2 ¿Sabe qué técnicas de ciencia de datos producen los mejores resultados?

El método crisp dm el cual implementa una metodología de prueba y error relacionado con la necesidad de negocio, es el que mejor da resultados debido a la lógica de mejora continua.

3.2.3 ¿Cómo se implementarán los resultados de modelado? ¿Ha considerado implementar su plan de proyecto?

Los resultados pueden ser implementados en una presentación como en un dashboard para la vista de los ejecutivos o mediante la creación de una API. Si el modelo está listo y produce un buen resultado en las predicciones de los tipos de cliente puede ser implementado mediante una la creación de una API, el cual se pondrá como un servicio en un servidor ya sea local o remoto en la nube, como en AWS, GCP o Azure.

3.2.4 ¿El plan de proyecto incluye todas las fases de CRISP-DM?

Solo las 3 primeras fases mas no las 3 últimas ya que nuestro estudio en la materia abarcó solo estas 3 primeras etapas. En las siguientes clases o siguientes módulos probablemente seamos capaces de incluir las siguientes 3 fases.

Fase de entendimiento del negocio: Realizar préstamos a personas que demostraron ser confiables y tener precaución con aquellos que no para evitar pérdidas financieras.

Restricciones: El algoritmo no categoriza a los clientes por distritos ya que pueden existir distritos con múltiples tipos de personas.

Estrategia preliminar: Se categoriza a los clientes por sus números de cuentas para obtener el número de transacciones realizadas, el monto total de dinero que mueve cada cliente y la frecuencia de uso de cuenta de cada cliente, ya que estos parámetros nos indican la posibilidad de un cliente a pagar su cuenta (Mientras más dinero mueve, mayor probabilidad de generar ganancia tiene el cliente)

Objetivo: Obtener los tipos de clientes que son buenos candidatos para otorgarle una tarjeta o un préstamo. Si continuamos con el desarrollo del modelo de Machine Learning si incluirá todas las últimas 3 fases de CRISP-DM , ya que en aquí primero se partió por identificar el caso o una necesidad de negocio, cargar los datos mediante la herramienta pentaho, transformar y unir los datos con sentencias de SQL y entender los datos que se puede utilizar gráficos en excel o con pandas de python, después seguiría el modelado, la evaluación de varios modelos entrenados mediante un score relacionado también con el negocio y por último su implementación en un dashboard que consuma una API la cual corre en un servidor local o remoto en la nube.

3.2.5 ¿Los riesgos y dependencias se incluyen en el plan?

Los riesgos para llevar a cabo el plan son:

Riesgos técnicos: Puede haber riesgos relacionados con la disponibilidad y confiabilidad de la infraestructura tecnológica necesaria para el proyecto, como problemas de manejo de diferentes lenguajes o softwares de análisis de datos, así como también tendríamos que contar con equipo de cómputo necesario, ya que no es lo mismo analizar 100 mb de datos a 100 gb de datos en donde se requeriría utiliza tecnologías como spark para procesar los datos.

Riesgos de datos: Los datos utilizados en el proyecto pueden contener errores, inconsistencias como valores nulos o estar incompletos, lo que puede afectar la calidad de los resultados y las conclusiones obtenidas.

Riesgos de recursos: Puede haber riesgos relacionados con la asignación de recursos humanos, financieros y de tiempo necesarios para llevar a cabo el proyecto ya que si es que no se hace una buena planificación sobre el uso de dinero provocaría retrasos en la finalización del proyecto de corto plazo, pasarían a largo plazo o simplemente ya no se ejecutaría.

Riesgos sobre el alcance: Existe el riesgo de que el alcance del proyecto no se defina adecuadamente desde el principio, lo que puede llevar a cambios y retrasos significativos a medida que se desarrolla el proyecto.

Riesgos de seguridad y privacidad: Cuando se trabajan con datos sensibles, existe el riesgo de violaciones de seguridad o de privacidad, lo que puede tener consecuencias legales y reputacionales, por lo cual necesitan ser anonimizados, en este proyecto se anonimizo los datos.

Las dependencias del plan para realizar el proyecto serían:

Dependencias de datos: El proyecto depende de la disponibilidad y acceso a conjuntos de datos, ya que muchas veces se necesitan permisos para acceder a ellos y muchas veces están anonimizados las columnas con información sensible. Aparte de la disponibilidad se depende de la calidad, la granularidad y la integridad de los datos utilizados, ya que habrían datos duplicados, nulos, valores faltantes,etc.

Dependencias de recursos: como los humanos, financieros o tecnológicos necesarios para llevar a cabo el proyecto. Por ejemplo, el proyecto puede requerir habilidades del manejo de lenguajes de JAVA, SCALA, PYTHON o conocimientos especializados que no estén disponibles internamente como APACHE SPARK, KAFKA,etc y deban adquirirse o contratar personas externas al equipo original. Dependencias de tiempo: los proyectos

muchas veces dependen de entregables, lo que significa que ciertas actividades sólo pueden comenzar una vez que se hayan completado otras. No se puede modelar e implementar un modelo sin haber definido el pipeline de ETL o ELT para la ingesta de datos. Pueden ser proyectos de corto plazo como de 1 año y de largo plazo como 3 años.

4 ENTENDIMIENTO DE LOS DATOS

4.1 Univariado con todas las variables que componen el dataset realizar lo siguiente:

- Clasificación de las variables entre variables cualitativas (ordinales o nominales) y cuantitativas (discretas o continuas).

Clasificación		Tipo_cliente	cualitativa - nominal
frecuencia	cualitativa - ordinal	cantidad_transacciones	cuantitativa - discreta
antg_cuenta	cuantitativa - continua	credito	cuantitativa - continua
distrito	cualitativa - nominal	retiro	cuantitativa - continua
region	cualitativa - nominal	trasferencia de fondos	cuantitativa - continua
habitantes	cuantitativa - discreta	trim_1_1993	cuantitativa - continua
num_ciudades	cuantitativa - discreta
ratio_residentes_urbanos	cuantitativa - continua	trim_4_1998	cuantitativa - continua
Salario_promedio	cuantitativa - discreta	intereses abonados	cuantitativa - continua
ratio_desempleo_prom	cuantitativa - continua	intereses por sanción si el saldo es negativo	cuantitativa - continua
crimenes_prom	cuantitativa - continua	credito en efectivo	cuantitativa - continua
n_empresarios_por_1000hab	cuantitativa - discreta	pago del credito/prestamo	cuantitativa - continua
tipo_tarjeta	cualitativa - ordinal	pago del seguro	cuantitativa - continua
antg_anios	cuantitativa - continua	pago por emision del extracto	cuantitativa - continua
fecha_prestamo	fecha	pagos domesticos	cuantitativa - continua
monto_prestamo	cuantitativa - continua	pension de vejez	cuantitativa - continua
duracion_prestamo	cuantitativa - continua	monto_transac_prom	cuantitativa - continua
pagos_mensuales	cuantitativa - discreta	balance_prom	cuantitativa - continua

- Obtener estadísticas de tendencia central, dispersión.

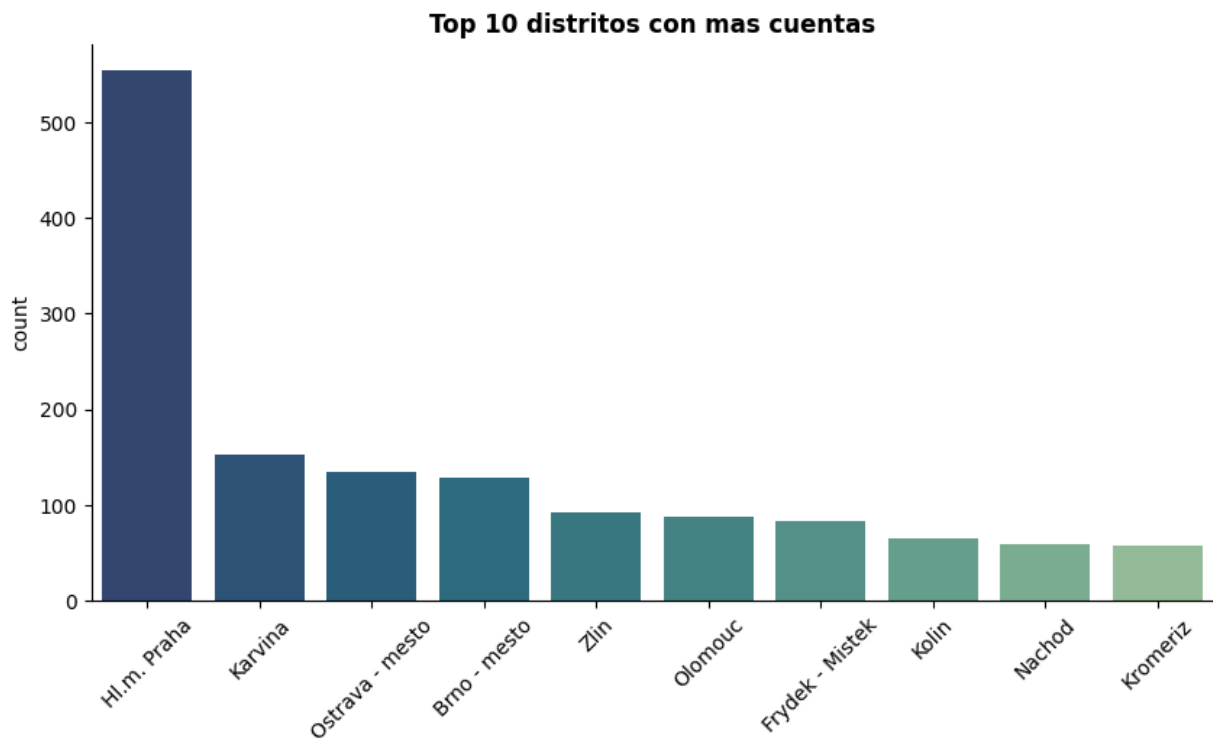
	count	mean	std	min	25%	50%	75%	max	rango_iqr	rango
antg_cuenta	4,500.00	3.40	1.51	1.01	2.17	3.00	5.02	6.00	2.85	4.99
habitantes	4,500.00	269,243.20	358,316.52	42,821.00	88,884.00	121,947.00	226,122.00	1,204,953.00	137,238.00	1,162,132.00
num_ciudades	4,500.00	5.53	2.91	1.00	4.00	6.00	8.00	11.00	4.00	10.00
ratio_residentes_urbanos	4,500.00	69.26	19.78	34.00	53.00	63.00	86.00	100.00	33.00	66.00
Salario_promedio	4,500.00	9,515.20	1,326.41	8,110.00	8,547.00	8,991.00	9,897.00	12,541.00	1,350.00	4,431.00
ratio_desempleo_prom	4,500.00	3.14	2.01	0.15	1.80	2.92	4.40	8.17	2.60	8.02
crimenes_prom	4,500.00	15,458.04	29,108.87	679.00	2,245.50	3,776.50	6,449.50	92,392.00	4,204.00	91,713.00
n_empresarios_por_1000hab	4,500.00	121.12	23.16	81.00	105.00	116.00	131.00	167.00	26.00	86.00
antg_anios	892.00	1.28	1.10	0.01	0.41	0.99	1.93	5.15	1.53	5.15

monto_prestamo	682.00	151,410.18	113,372.41	4,980.00	66,732.00	116,928.00	210,654.00	590,820.00	143,922.00	585,840.00
duracion_prestamo	682.00	36.49	17.08	12.00	24.00	36.00	48.00	60.00	24.00	48.00
pagos_mensuales	682.00	4,190.66	2,215.83	304.00	2,477.00	3,934.00	5,813.50	9,910.00	3,336.50	9,606.00
cantidad_transacciones	4,500.00	234.74	126.85	9.00	133.00	208.00	330.00	675.00	197.00	666.00
credito	4,500.00	717,216.06	670,245.69	19,700.00	228,050.57	466,683.34	1,005,418.66	3,857,257.50	777,368.05	3,837,557.50
retiro	4,500.00	46,356.39	103,912.98	0.00	0.00	0.00	8,936.25	756,972.00	8,936.25	756,972.00
trasferencia de fondos	4,500.00	627,048.31	584,528.81	1,400.00	197,326.10	413,341.50	883,291.92	3,590,676.00	685,965.84	3,589,276.00
trim_1_1993	298.00	57,237.52	89,182.79	200.00	4,045.75	27,497.85	83,355.90	801,600.06	79,310.15	801,400.06
trim_2_1993	554.00	165,741.92	198,368.02	200.00	27,512.45	100,470.45	244,377.16	1,421,449.88	216,864.71	1,421,249.88
trim_3_1993	858.00	223,371.83	240,611.12	200.00	45,397.62	156,683.03	306,884.81	1,723,478.38	261,487.19	1,723,278.38
trim_4_1993	1,139.00	274,861.41	250,697.19	200.00	79,709.30	225,303.09	405,944.34	1,437,585.50	326,235.05	1,437,385.50
trim_1_1994	1,229.00	327,113.09	229,713.12	200.00	167,495.70	279,144.47	450,729.31	1,480,876.50	283,233.61	1,480,676.50
trim_2_1994	1,360.00	336,051.78	242,239.53	200.00	168,257.25	282,597.69	443,683.19	1,649,813.75	275,425.94	1,649,613.75
trim_3_1994	1,468.00	325,001.94	238,823.83	200.00	169,275.28	278,751.12	410,994.84	1,733,188.75	241,719.56	1,732,988.75
trim_4_1994	1,578.00	362,844.22	250,937.94	200.00	201,614.48	309,475.62	472,924.53	2,177,462.50	271,310.05	2,177,262.50

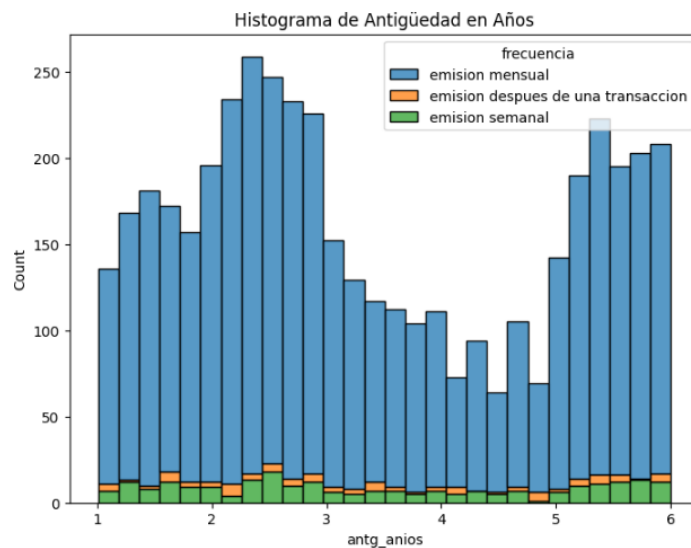
trim_1_1995	1,734.00	365,394.75	247,179.34	200.00	207,764.84	309,731.25	494,490.53	1,724,001.62	286,725.69	1,723,801.62
trim_2_1995	1,893.00	348,609.38	254,640.17	200.00	177,567.59	287,501.31	455,825.47	1,755,906.38	278,257.88	1,755,706.38
trim_3_1995	2,059.00	334,694.50	245,313.05	200.00	173,155.59	283,488.31	442,711.50	2,153,078.25	269,555.91	2,152,878.25
trim_4_1995	2,239.00	367,193.50	258,147.31	-8,312.40	189,663.34	315,366.69	488,954.50	1,844,082.12	299,291.16	1,852,394.53
trim_1_1996	2,573.00	358,774.94	271,706.50	-7,228.80	175,416.25	300,051.94	501,092.03	2,088,404.38	325,675.78	2,095,633.17
trim_2_1996	2,878.00	333,728.16	249,532.42	-10,579.65	158,392.11	278,085.44	461,212.78	1,615,236.38	302,820.67	1,625,816.03
trim_3_1996	3,256.00	324,543.81	254,778.73	-3,340.80	149,885.27	264,821.69	435,042.38	2,338,191.75	285,157.11	2,341,532.55
trim_4_1996	3,598.00	363,609.94	278,769.22	-4,725.45	167,831.56	301,944.28	493,815.84	2,505,911.50	325,984.28	2,510,636.95
trim_1_1997	3,806.00	389,516.38	276,065.94	200.00	188,420.52	331,316.72	533,503.25	1,812,046.50	345,082.73	1,811,846.50
trim_2_1997	4,044.00	374,149.94	269,819.22	-6,940.20	177,156.45	312,780.12	505,343.81	1,884,667.25	328,187.36	1,891,607.45
trim_3_1997	4,289.00	362,923.34	272,383.25	-12,693.10	170,075.09	301,371.94	479,526.41	2,119,564.00	309,451.31	2,132,257.10
trim_4_1997	4,492.00	404,443.97	290,929.25	-34,224.45	191,068.22	336,788.31	542,133.50	2,222,533.75	351,065.28	2,256,758.20
trim_1_1998	4,487.00	437,812.38	291,556.47	-57,899.80	215,124.45	368,715.50	592,302.88	1,987,310.38	377,178.42	2,045,210.18
trim_2_1998	4,482.00	416,489.94	287,594.16	-98,279.35	198,020.50	346,767.38	559,076.44	1,964,130.12	361,055.94	2,062,409.48
trim_3_1998	4,484.00	402,883.19	286,025.19	-122,003.75	186,683.25	330,881.38	532,163.12	1,957,924.00	345,479.88	2,079,927.75
trim_4_1998	4,484.00	421,596.72	296,836.25	-80,420.50	194,858.64	346,434.56	568,667.75	2,148,081.75	373,809.11	2,228,502.25

intereses abonados	4,500.00	6,104.60	4,272.78	0.00	2,971.55	5,055.65	8,168.27	31,208.90	5,196.72	31,208.90
intereses por sanción si el saldo es negativo	4,500.00	8.37	98.59	0.00	0.00	0.00	0.00	3,083.70	0.00	3,083.70
credito en efectivo	4,500.00	1,218,265.06	1,288,804.54	11,890.00	236,572.50	777,399.00	1,754,293.25	7,384,295.00	1,517,720.75	7,372,405.00
pago del credito/prestamo	4,500.00	12,278.29	39,466.68	0.00	0.00	0.00	0.00	412,850.00	0.00	412,850.00
pago del seguro	4,500.00	5,372.27	30,238.20	0.00	0.00	0.00	0.00	650,208.00	0.00	650,208.00
pago por emsion del extracto	4,500.00	596.98	595.23	0.00	292.00	467.20	817.60	6,900.00	525.60	6,900.00
pagos domesticos	4,500.00	110,779.20	134,595.17	0.00	0.00	68,142.00	161,335.50	957,700.00	161,335.50	957,700.00
pension de vejez	4,500.00	37,216.03	94,904.47	0.00	0.00	0.00	0.00	485,380.00	0.00	485,380.00
monto_transac_prom	4,500.00	5,757.69	4,030.53	843.44	2,303.21	4,729.72	8,352.16	21,062.61	6,048.95	20,219.18
balance_prom	4,500.00	36,667.86	15,165.63	5,711.30	23,570.89	34,696.54	47,818.69	81,192.76	24,247.80	75,481.46

- Generacion gráficas dependiendo el tipo, histogramas, barras, boxplot. Las gráficas deben tener algún comentario o descubrimiento que puedan hallar.



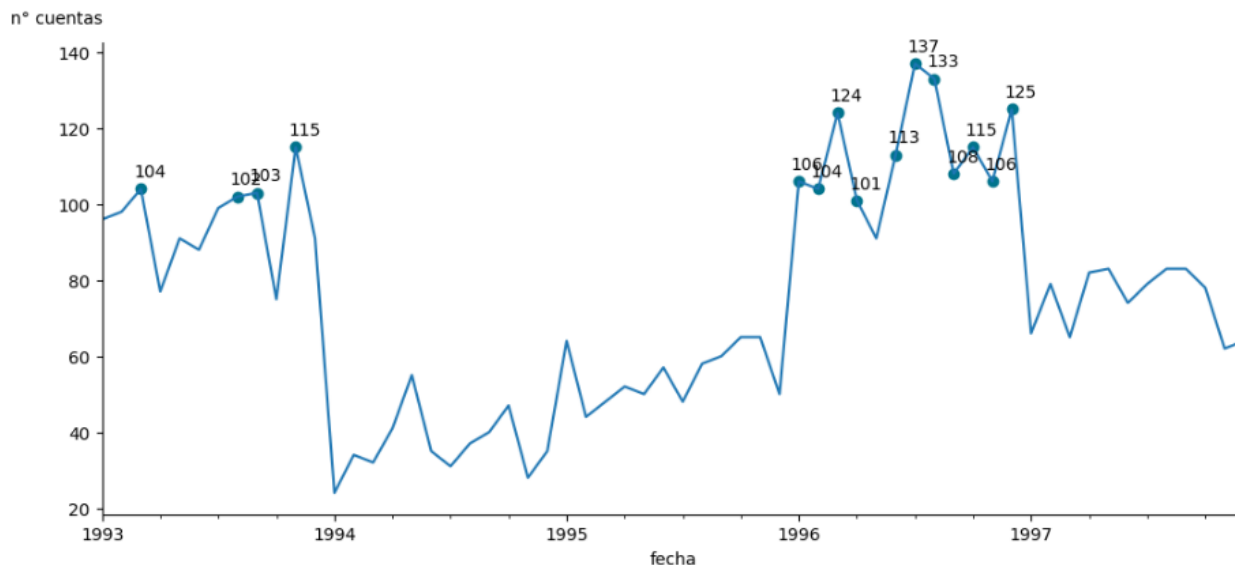
Se puede observar que entre todos los distritos el distrito Hl.m.Praha es el distrito con más clientes del banco Berka, el cual también es el único distrito en una región.



En el gráfico se observa un comportamiento aparentemente bimodal, entre las cuentas más antiguas (5-6) años y entre las cuentas seminuevas (2-3) años. Posiblemente exista un motivo para ese comportamiento de la apertura de la cuenta, por algún suceso económico, una mejora del servicio, tal vez mejoras de los intereses,etc.

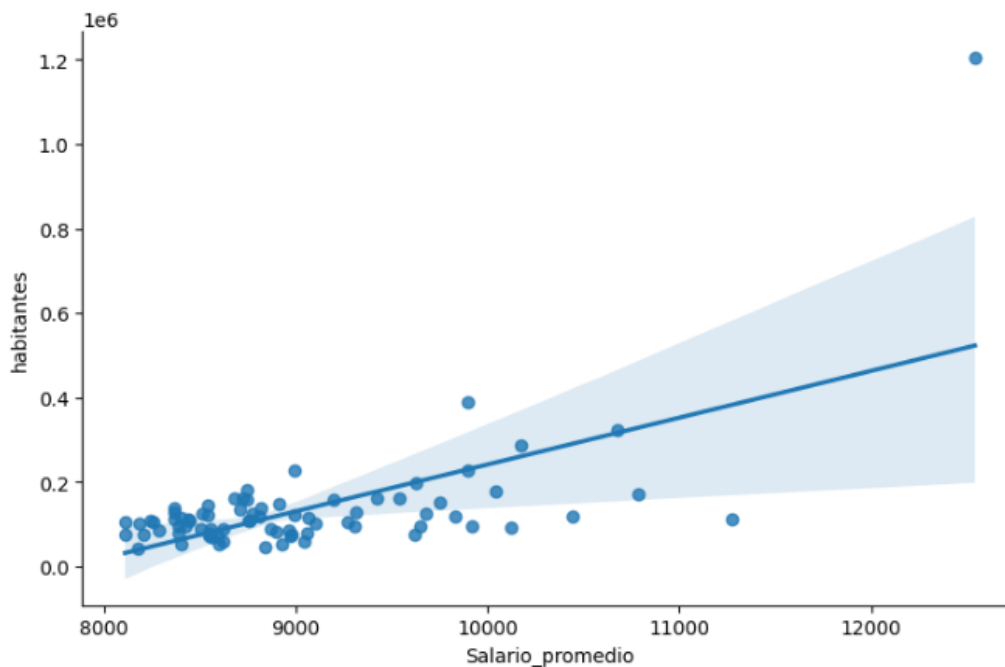
La mayoría de los clientes hacen una emisión mensual de sus saldos.

CUENTAS NUEVAS POR MES DESDE 1993 a 1997

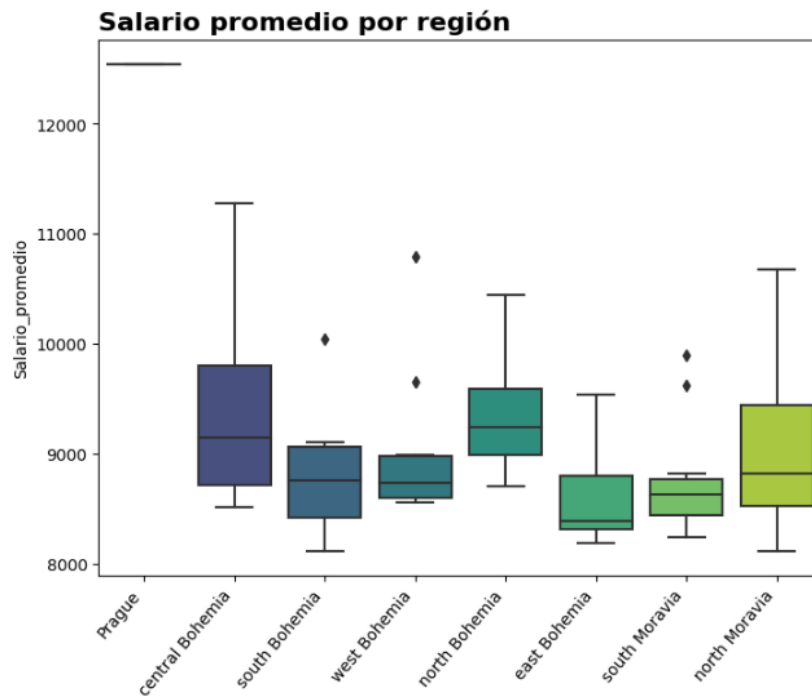


Este gráfico muestra la evolución mensual del número de nuevas cuentas entre los años 1993 hasta fines de 1997, se observa zonas de crecimiento y zonas bajas, las zonas más altas en crecimiento fueron durante los años de 1993 y 1996, siendo 137 nuevos clientes el número más alto para un mes.

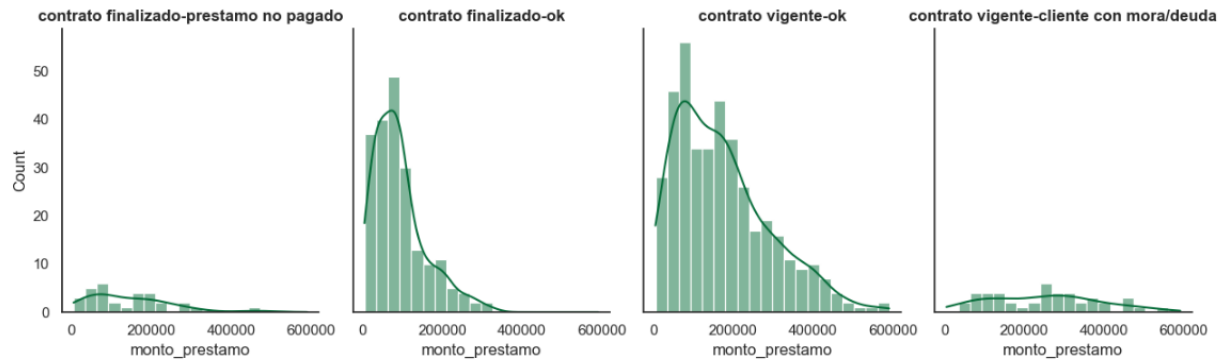
Salario promedio por numero de habitantes



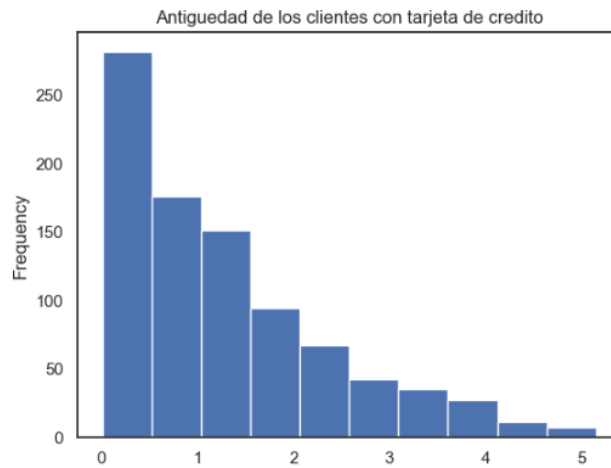
El gráfico muestra que existe una relación entre el salario promedio con el número de habitantes, a mayor número de habitantes por distrito mayor es el salario promedio.



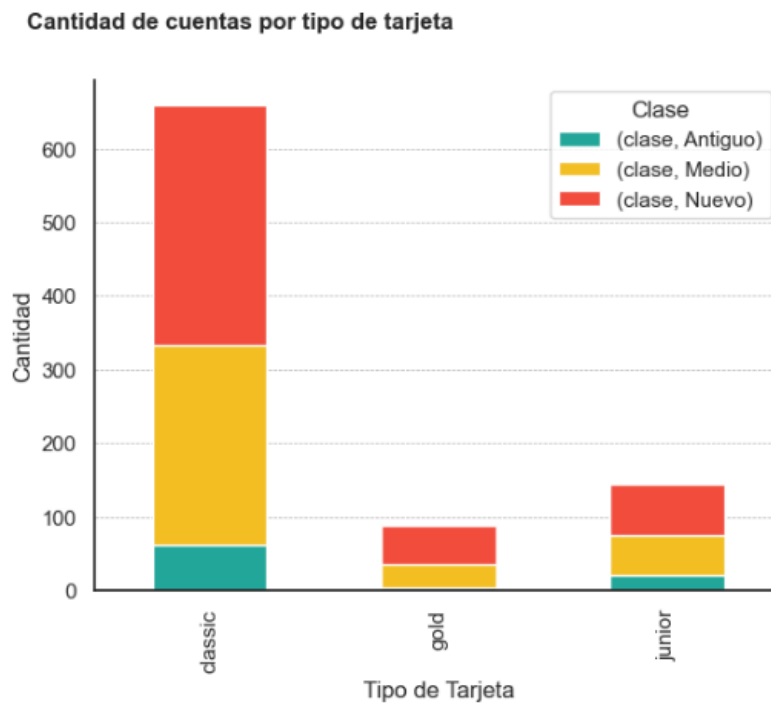
La región con el salario promedio más alto en base a la mediana, es la región de Prague que solo tiene un distrito llamado Hl.m. Praha y la región más baja en salario sería East Bohemia junto con South Moravia, el cual contiene 2 distritos que ganan más que el promedio de esa región.



En este gráfico se puede notar la distribución de los préstamos por su estado, y se observa que muchos clientes solicitan préstamos entre el rango de 0 a 200 000 \$ y la terminan pagando pero algunos otros no y habrá que asociarlos con otras variables.

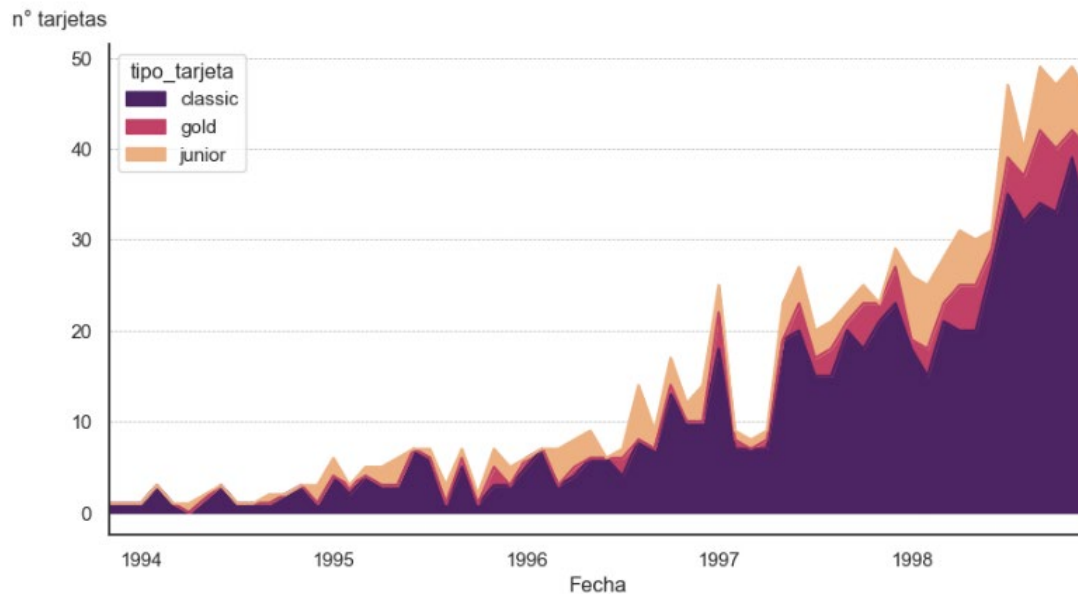


- Distribución de la antigüedad de los clientes, se tienen muchos clientes que son nuevos y seminuevos (< 2 años) y pocos clientes antiguos que poseen su tarjeta de crédito.

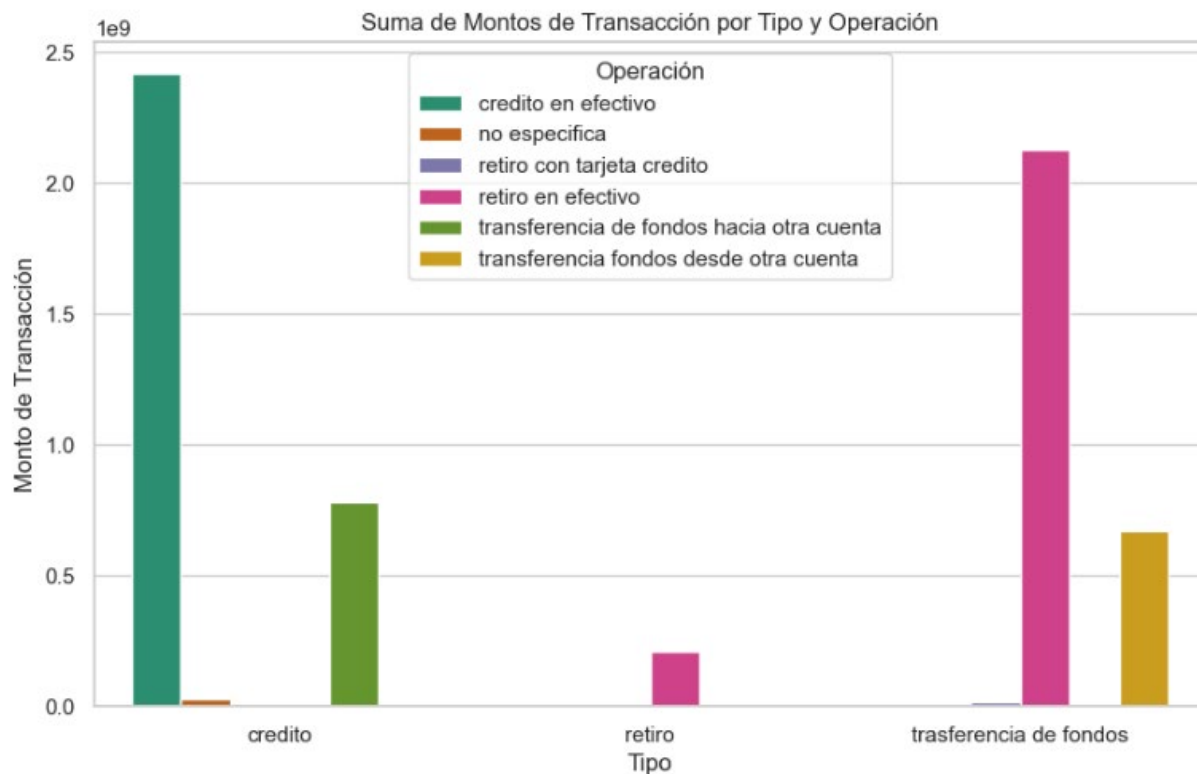


En el gráfico se observa la cantidad de cuentas por tipo de tarjeta agrupadas por la clase de cuenta (antiguo, medio, nuevo), donde se observa que la mayoría de los clientes (73.88%) poseen una tarjeta clásica y en menor medida una tarjeta junior o gold.

Cantidad de Tarjetas dadas por mes desde 1993 a 1998



En el gráfico se observa la evolución de la cantidad de tarjetas otorgadas de manera mensual y la proporción por tipo de tarjeta de crédito. Se ve también que existió una tendencia creciente al pasar los años, en especial para las tarjetas clásicas.



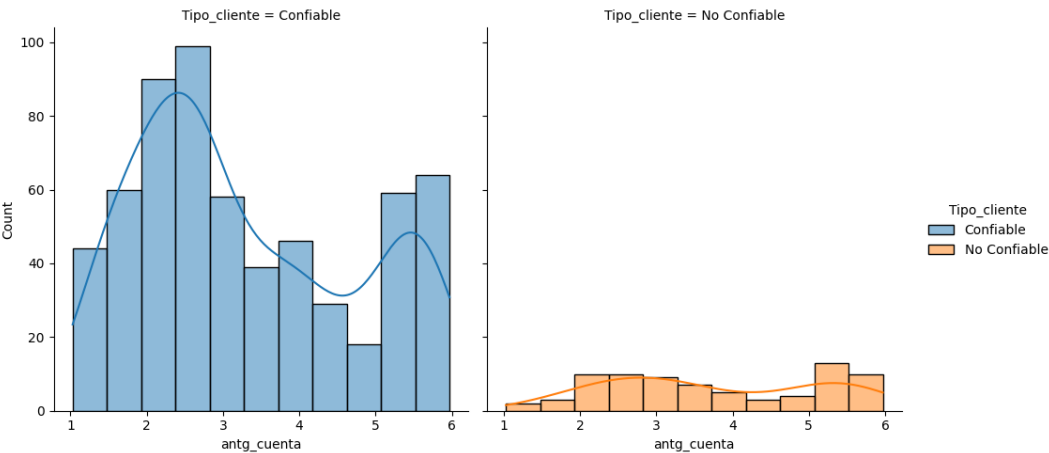
En el gráfico se observa la evolución de la cantidad de tarjetas otorgadas de manera mensual y la proporción por tipo de tarjeta de crédito. Se ve también que existió una tendencia creciente al pasar los años, en especial para las tarjetas clásicas.

4.2 Bivariado con las variables que ingresen en el estudio, respecto de la variable objetivo o target.

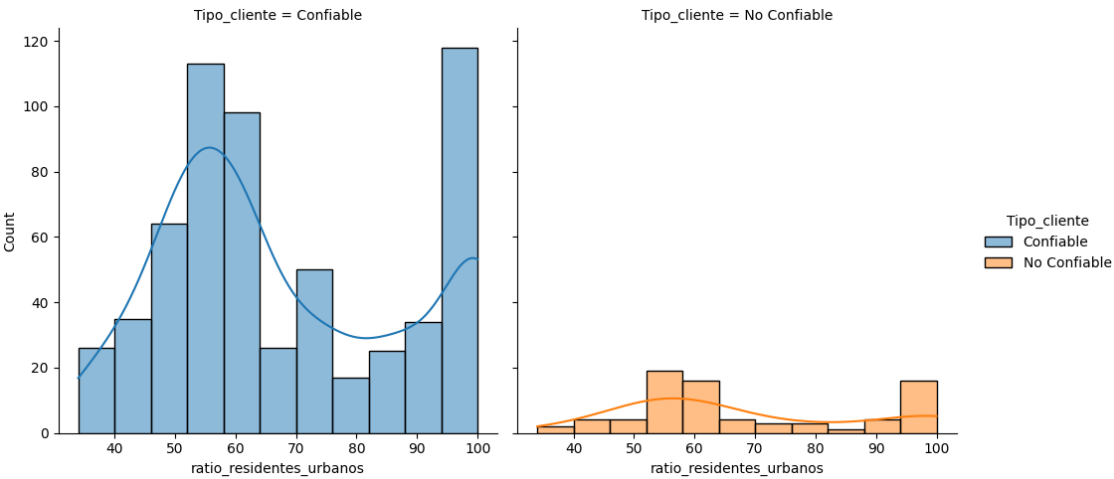
	monto_prestamo	duracion_prestamo	pagos_mensuales
estado			
contrato finalizado-ok	91,641.46	22.23	4,264.14
contrato finalizado-prestamo no pagado	140,720.90	25.55	5,396.26
contrato vigente-cliente con mora/deuda	249,284.53	46.13	5,286.64
contrato vigente-ok	171,410.35	43.44	3,938.54

Las personas que se retrasa en pagar los interés son los que adquirieron en promedio un préstamo elevado, alrededor de \$ 249 284, los cuales presentan una duración más larga a las demás y con pagos mensuales más altos.

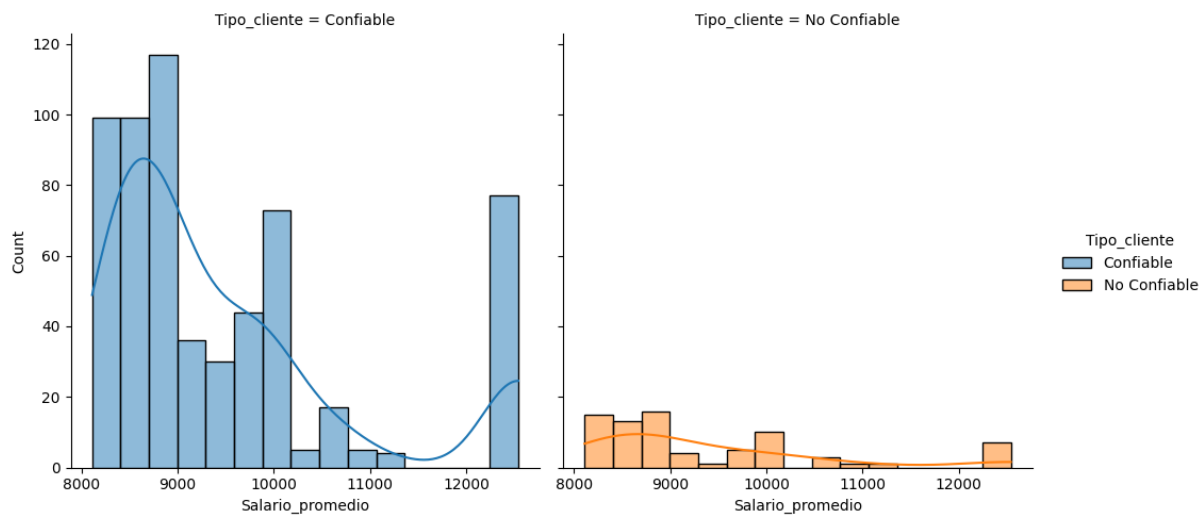
Las personas que no tienen problemas en pagar su deuda, son las que solicitan un préstamo alrededor de \$ 90 000, las cuales tiene una duración menor.



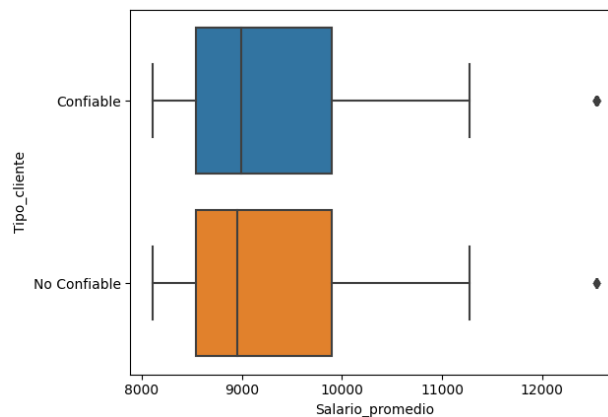
Se puede observar que los clientes que no pagan sus deudas tiende a hacer de cuentas seminuevas en cierta parte.



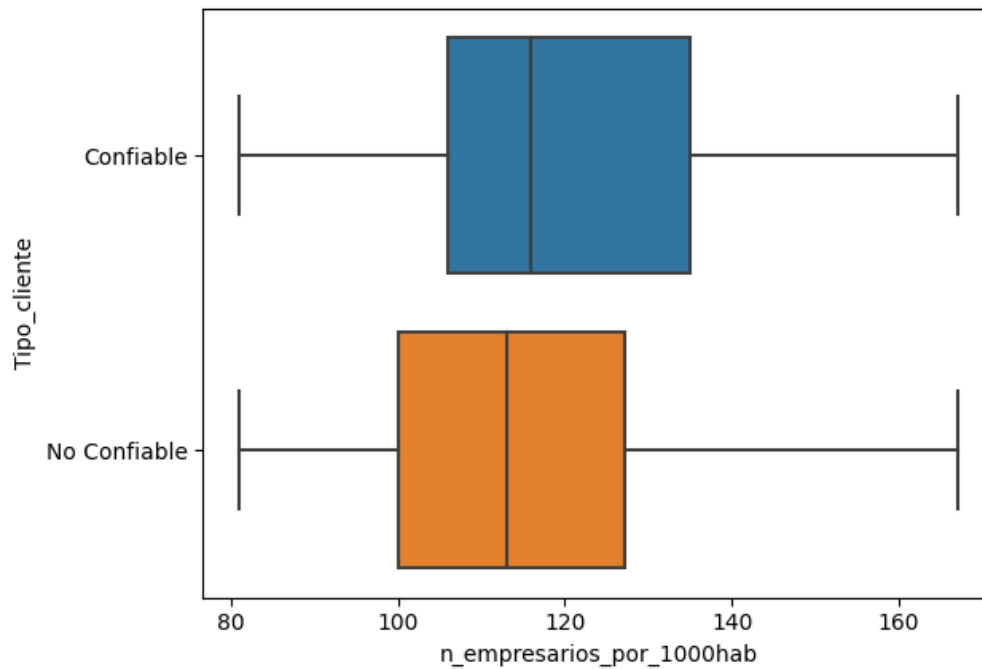
Se observa que no hay mucha diferencia comparado con porcentaje urbano de donde viven los clientes, por lo cual los clientes provienen de ciudades urbanas similares.



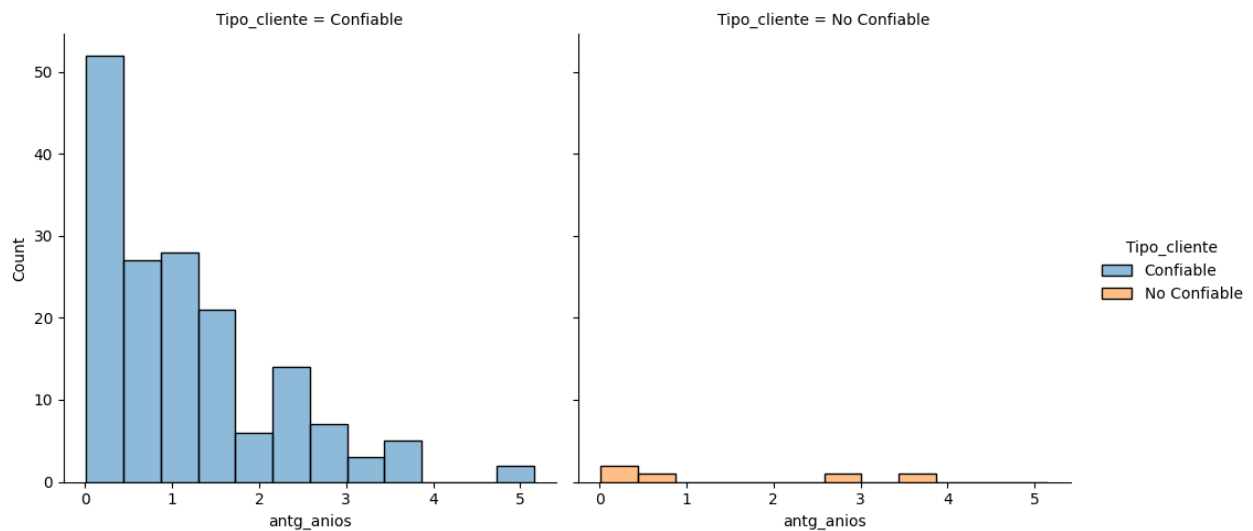
Para complementar este gráfico se realizó una comparación mediante los boxplots, el cual es el siguiente gráfico.



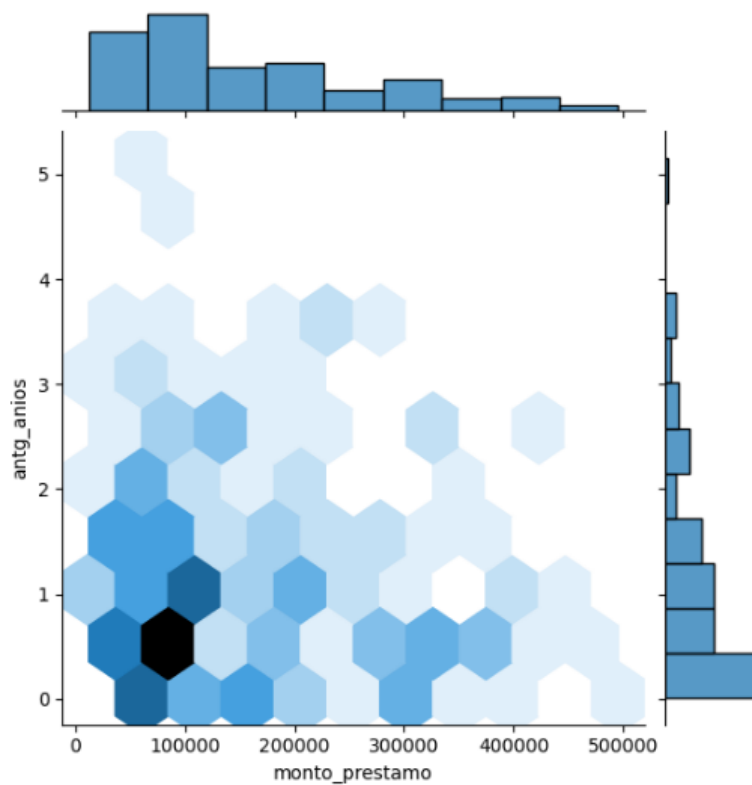
Se observa que el salario promedio no está muy distante entre los que pagan y los que no. De hecho es casi igual. Nosotros pensábamos que no pagaban los que tenían salario promedio más bajo pero el anterior gráfico desmiente nuestra hipótesis.



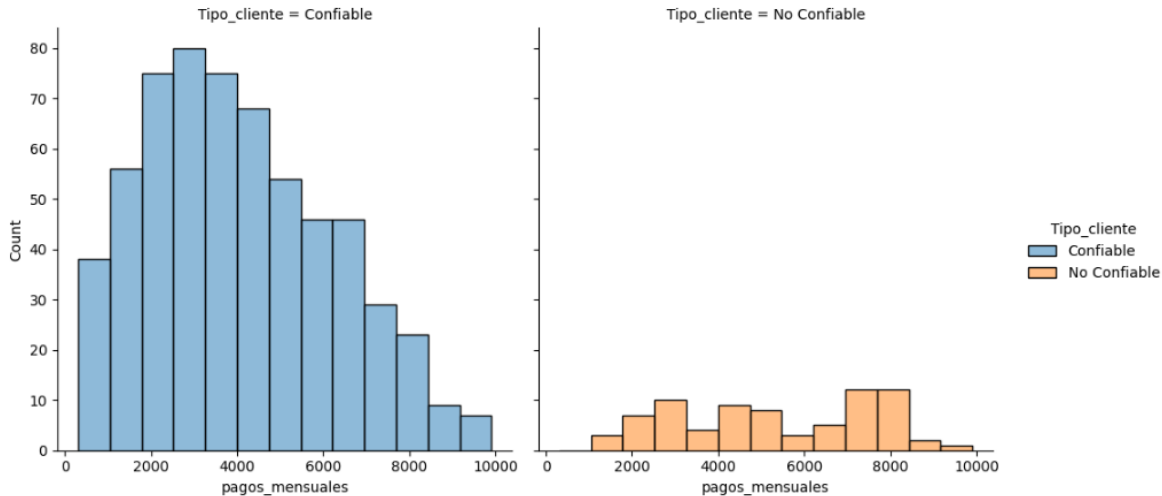
Se observa que con respecto a la mediana, los que no pagan sus deudas viven en lugares donde el emprendimiento es un poco más bajo.

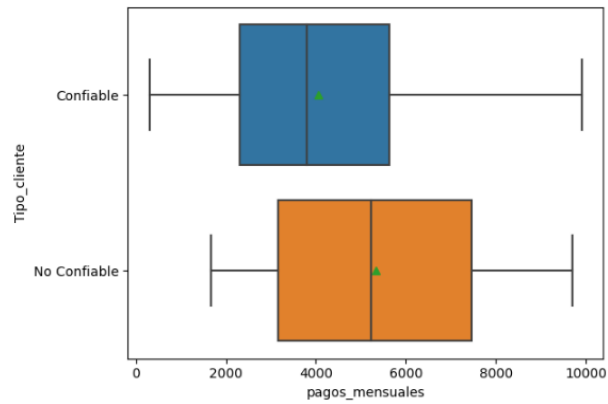


Se puede observar que hay mucha diferencia entre la antigüedad de la tarjeta, no se ve alguna tendencia clara de los que no pagan, lo que sí se puede ver es la tendencia de los que sí pagan, se ve que un sesgo hacia la derecha, lo cual indica que hay muchas cuentas nuevas que piden créditos, pero también hay nuevos que no pagan sus créditos.

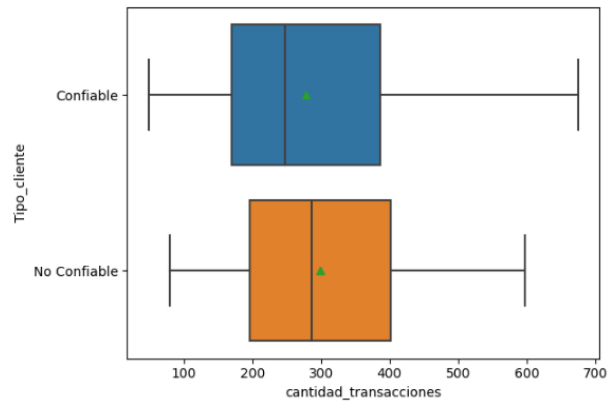


Se puede observar que las cuentas nuevas con tarjeta, solicitan muchos préstamos, pero no muy elevados a diferencia de las cuentas antiguas que solicitan pocas.

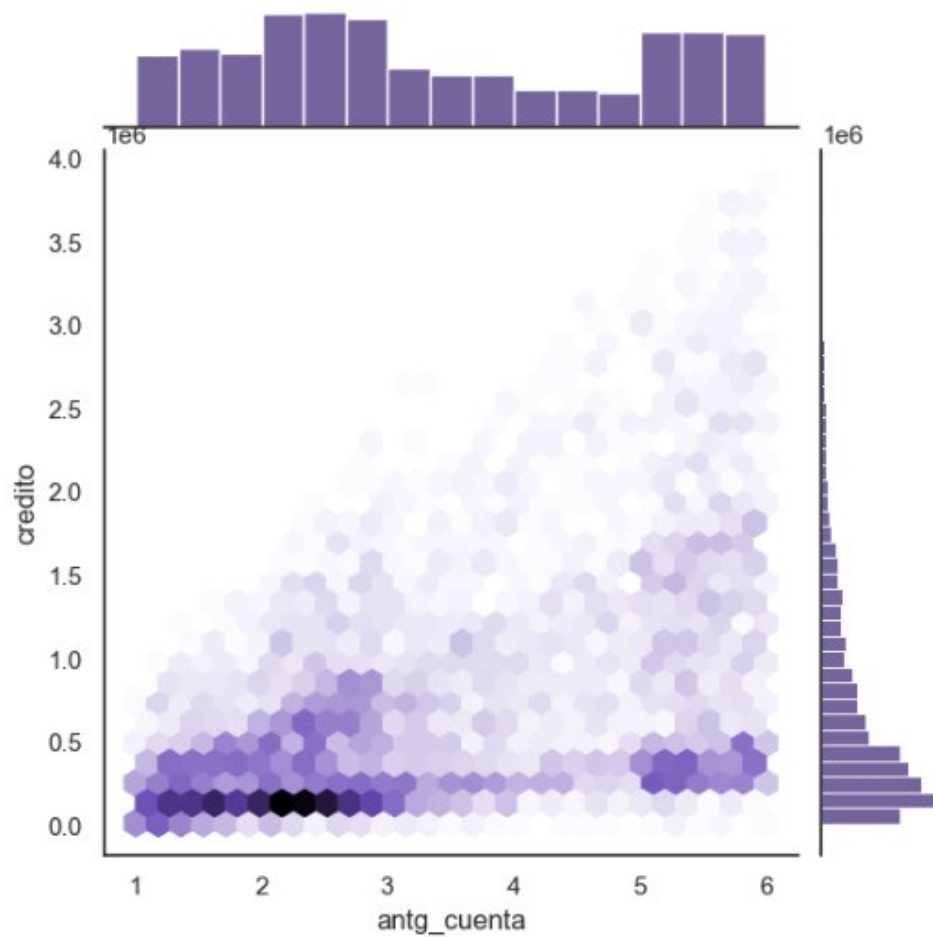




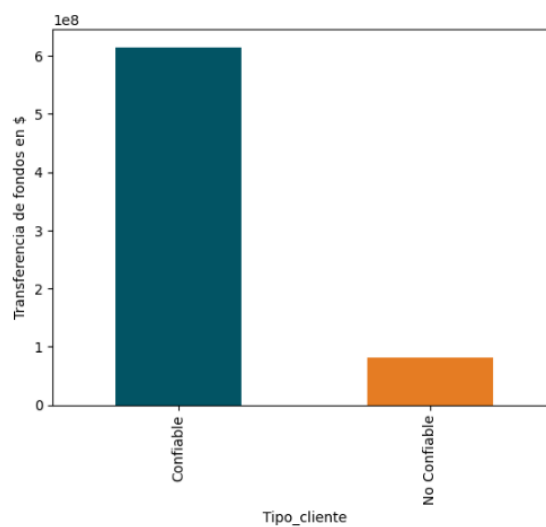
Aquí sí se puede ver una diferencia interesante, los que no pagan tienden a realizar pagos mensuales mayores que los que pagan. Al ser montos más elevados tal vez no se evaluó correctamente su rango crediticio.



Se puede observar que con respecto a la mediana, los que clientes que no pagan tienden a hacer más transacciones que los que sí pagan su deuda.

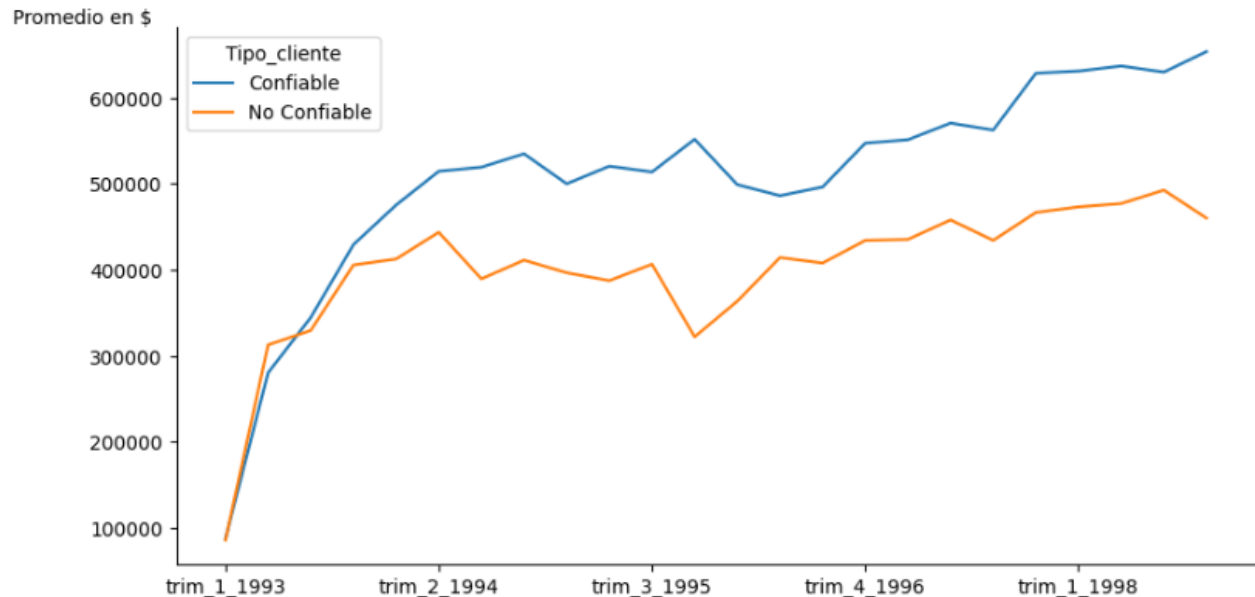


- También se puede observar la relación entre las dos variables antigüedad de la cuenta y el monto total del crédito solicitado. Se puede observar que puede existir una relación lineal entre las dos variables, habría una cierta tendencia de a mayor antigüedad se solicitan montos grandes de préstamos pero en poca cantidad, mientras que las cuentas nuevas solicitan pocos créditos pero en mayor cantidad.



Se observa que los clientes que realizan menos transferencias son clientes no confiables, es decir que no pagan su deuda. Pero esto puede deberse a que los datos están desbalanceados con respecto al estado hay más clientes confiables etiquetados que los que no son confiables o no pagaron su deuda.

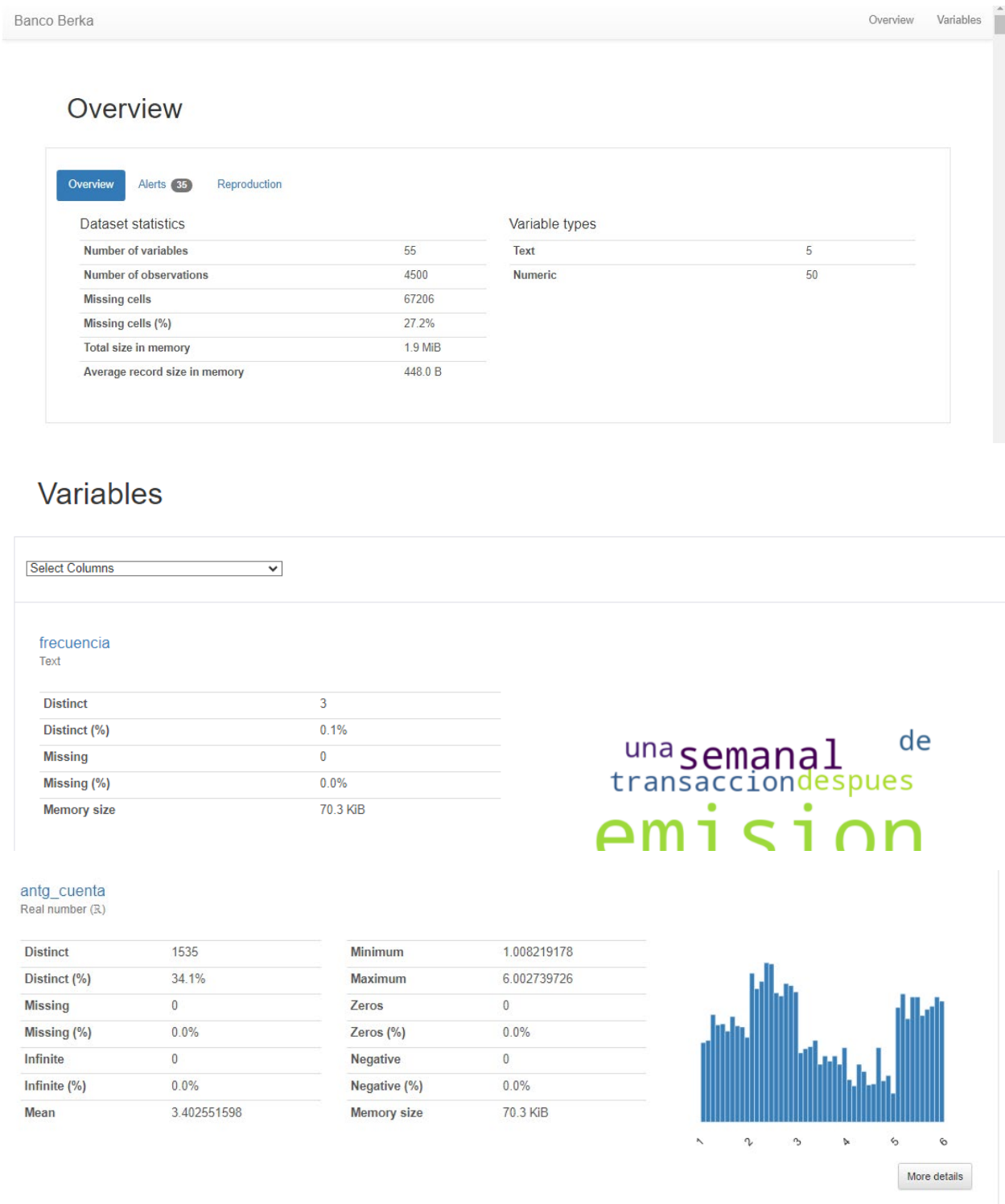
Promedio Transaccion-Balance desde 1993 a 1998



- Este gráfico se creó en base a los promedios entre la transacción y el balance para cada cuenta.
- Se observa un patrón claro en este gráfico y una diferencia entre los que no pagan y los que sí pagan sus deudas, se ve que los que pagan sus deudas tienen un promedio balance-transacción mejor que los que no pagan sus deudas.
- Este comportamiento puede servirnos para realizar comparaciones con cada cliente y ver a qué cliente se acercan más y tal vez poder otorgar o ofrecer un préstamo a un cliente o cuenta correctamente, e ir seleccionando a las cuentas para ofrecer un servicio o descuento.
- Sería importante considerar estas variables trimestrales para desarrollar un algoritmo de aprendizaje supervisado para clasificación de las demás cuentas.

4.3 Generación un perfilado del total de variables que **usará** para la construcción del modelo.

El perfilado se hizo con la librería Pandas Profiling.



4.3.4 ¿Le han ayudado estos atributos a formular hipótesis?

Si, por ejemplo las personas con un mayor balance promedio tienden a pagar su deuda y las otras no, por lo que podríamos pensar que esta variable tiene mucha importancia en la predicción futura para otorgar un crédito.

4.3.5 ¿Ha detectado el tamaño de todos los orígenes de datos?

si, se utilizaron comandos en pandas para observar el tamaño de cada tabla, como el comando `df.shape`. así como también sus tipos de datos de cada variable.

4.3.6 ¿Puede utilizar un subconjunto de datos cuando lo estime conveniente?

Si se podría hacer un muestreo a nuestros para estimar valores poblacionales de los clientes debido a que existe una buena cantidad de datos aunque seria mejor si tuviéramos registros de otros años más recientes, para ver el panorama casi actual del banco en relación a los créditos.

4.3.7 ¿Ha calculado los estadísticos básicos de cada atributo de su interés? ¿Ha obtenido información de interés?

Si, se ha calculado utilizando una función propia que agrega el rango y el rango intercuartil a la función de pandas `.describe` con el objeto de tener una mejor vista de todas las columnas numéricas. Con ello obtuvimos estadísticos importantes para ver los valores promedios de cada variable y su variabilidad respecto a ese promedio con la desviación estándar.

4.3.8 ¿Ha utilizado gráficos de exploración para obtener atributos clave? ¿Este conocimiento ha reformulado alguna de sus hipótesis?

Se usó gráficos de relación como los scatterplots para ver la relación con otras variables y la variable objetivo. Se hicieron varias gráficas con la hipótesis de que pudieran haber diferencias por categorías, pero muchos de los gráficos nos mostraron que no existían diferencias entre ciertas características que creíamos que podría haber una distinción. Como por ejemplo en la antigüedad de los clientes, la cual no hubo mucha diferencia en la distribución de los que pagan su deuda y de los que no.

4.3.9 ¿Cuáles fueron los problemas de calidad de datos del proyecto? ¿Tiene una planificación para resolver estos problemas?

se tuvo las siguiente planificación respecto a la revisiones de los datos como:

Problema: Datos incompletos o faltantes en la base de datos.

Plan: Explorar por que faltan y si tienen o no sentido que falten, en este caso si había datos faltantes en una tabla, que fue en transacciones, se exploró y se vio que era lógico la falta ya que en un campo el banco otorga créditos en efectivo, por lo cual no había una cuenta de la provenia el crédito por que el banco lo otorga ya sea por su tarjeta o préstamo. Lo mismo con los retiros en efectivo, era lógico que falte datos innecesarios para ese tipo de transacciones, ya que no debía tener banco de destino, ni cuenta de donde proviene.

Problema: Datos inconsistentes en la estructura/formato, tipos incorrectos. Se detectó que algunas variables tipo numéricas eran tratadas como objetos, lo cual impide realizar operaciones, como el de sacar promedios.

Plan: Detectar mediante código en pandas y convertir a los tipos correctos mediante pandas.

Problema: Datos duplicados, se repetían datos en algunos joins.

Plan: Entender las relaciones que tienen entre cada tabla 1:1 o 1:muchos y así podemos saber qué esperar de los joins hechos. Revisar que tipo de join se hace, ya que son diferentes. Revisar los tamaños al final de cada join para ver si el tamaño es consistente o lógico con el join realizado.

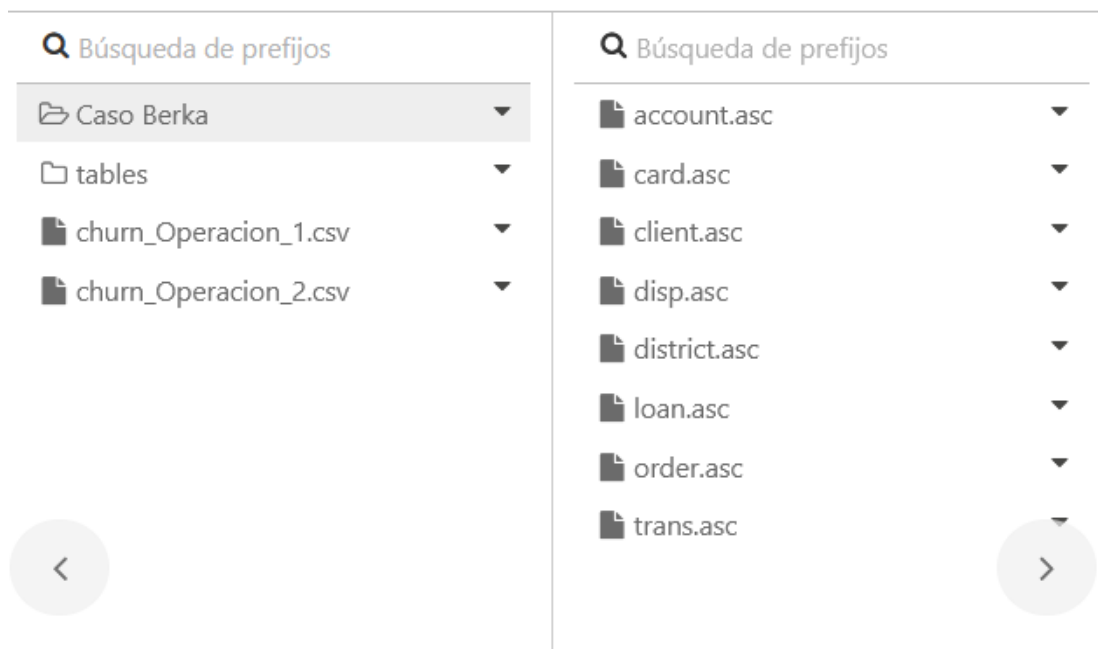
4.3.10 ¿Las fases de preparación de los datos son claras? Por ejemplo, ¿sabe qué orígenes de datos debe fusionar y los atributos que debe filtrar o seleccionar?

Si son claras ya que se realizó el armado del dataset a través de una serie de pasos de transformación y corrección de tipos de datos, posteriormente se realizó el join de cada tabla transformada paso a paso. Jupyter permite documentar cada paso, por lo cual se hizo un comentado de las líneas de código importantes. Al igual que se estableció que todo se ejecute de manera lineal y no con saltos, celda por celda en Jupyter notebook, esto para evitar confusiones entre el equipo al momento de compartir un notebook.

5 Preprocesamiento de los datos

5.1 Manejo de datos con DataBricks

Subimos todos los archivos al DBFS



5.1.1 Crear tablas

Con el código siguiente debemos crear todas las tablas necesarias para trabajar con el caso Berka

Preprocesamiento de Datos

```
1 %sql
2 DROP TABLE IF EXISTS account;
3 CREATE TABLE account
4 USING csv
5 OPTIONS (path "/FileStore/Caso Berka/account.asc",delimiter ";", header "true");
6
7 DROP TABLE IF EXISTS card;
8 CREATE TABLE card
9 USING csv
10 OPTIONS (path "/FileStore/Caso Berka/card.asc",delimiter ";", header "true");
11
```

Obteniendo todas las tablas necesarias que podemos verificar con el siguiente comando:

```
1 %sql
2 show tables;
```

► _sqldf: pyspark.sql.dataframe.DataFrame = [database: string, tabl

Tabla ▾ +

	database	tableName	isTemporary
1	default	account	false
2	default	aux_telco_operacion1f	false
3	default	card	false
4	default	client	false
5	default	disp	false
6	default	district	false
7	default	loan	false
8	default	orden	false
9	default	telco_operacion1	false
10	default	telco_operacion2	false

Revisamos cada tabla para ver si se subieron los datos correctamente:

```
1 %sql
2 select * from district
```

► (1) trabajos de Spark

► _sqldf: pyspark.sql.dataframe.DataFrame = [A1: string, A2: string ... 14 campos adicionales]

Tabla ▾ +

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
1	1	Hl.m. Praha	Prague	1204953	0	0	0	1	1	100.0	12541	0.29
2	2	Benesov	central Bohemia	88884	80	26	6	2	5	46.7	8507	1.67
3	3	Beroun	central Bohemia	75232	55	26	4	1	5	41.7	8980	1.95
4	4	Kladno	central Bohemia	149893	63	29	6	2	6	67.4	9753	4.64
5	5	Kolin	central Bohemia	95616	65	30	4	1	6	51.4	9307	3.85
6	6	Kutna Hora	central Bohemia	77963	60	23	4	2	4	51.5	8546	2.95
7	7	Mladá Boleslav	central Bohemia	84735	58	28	1	2	6	62.4	8870	2.96

77 filas | 0,40 segundos de tiempo de ejecución

Actualizado hace 1 hora

Aplicamos las transformaciones y Joins para obtener la tabla minable:

```

1 %sql
2 alter view transcompleta as select t.account_id as Cuenta, count(*) as Nro_movimientos,sum(t.amount) as Total_Dinero_movido,
3 CASE
4     WHEN l.status = "A" THEN "Excelente candidato"
5     WHEN l.status = "B" THEN "Inconfiable"
6     WHEN l.status = "C" THEN "Confiable"
7     WHEN l.status = "D" THEN "Dudoso"
8     else "No se presto"
9 end Confiabilidad,
10 CASE
11     WHEN a.frequency = "POPLATEK MESICNE" THEN "Uso mensual"
12     WHEN a.frequency = "POPLATEK TYDNE" THEN "Uso semanal"
13     when a.frequency = "POPLATEK PO OBRATU" THEN "Frecuente"
14 end Frecuencia, d.a3 as Region,d.a11 as Salario_Promedio,d.a14 as Empresarios_en_miles,d.a4 as Habitantes,
15 d.a10 as Urbanizacion,d.a13 as Desempleo,d.a16 as Crimenes
16 from trans t
17 inner join account a
18 on t.account_id =a.account_id
19 left join loan l
20 on l.account_id = a.account_id
21 left join district d
22 on d.a1 = a.district_id
23 group by t.account_id, Confiabilidad, Frecuencia, d.A3,d.a11,d.a14,
24 d.a4,d.a10,d.a13,d.a16;

```

Obteniendo la siguiente tabla:

Tabla

+

	Cuenta	Nro_movimientos	Total_Dinero_movido	Confiabilidad	Frecuencia	Region	Salario_Promedio	Empresarios_en_miles	Habitantes	Urban
1	813	392	1922343.0999999992	Excelente candidato	Uso mensual	south Bohemia	8427	107	93931	56.9
2	544	254	2191771.8999999994	Excelente candidato	Uso mensual	north Bohemia	9272	118	105058	81.0
3	9869	465	3044394.3	Confiable	Uso mensual	central Bohemia	8754	137	107870	58.0
4	2051	224	2893258.8	Dudoso	Uso semanal	south Moravia	9624	145	197099	74.7
5	7819	591	1298770.8999999994	Confiable	Uso mensual	north Bohemia	8965	104	85852	59.8
6	1843	462	2764436.8999999994	Excelente candidato	Uso mensual	central Bohemia	8754	137	107870	58.0
7	7753	500	6346623.200000005	Excelente candidato	Uso mensual	north Moravia	10673	100	323870	100.0
8	6118	408	3032913.2000000007	Inconfiable	Uso semanal	Prague	12541	167	1204953	100.0
9	2912	369	5610280.7	Confiable	Uso mensual	north Moravia	10673	100	323870	100.0
10	6555	438	4243596.7	Excelente candidato	Uso semanal	north Bohemia	8965	104	85852	59.8
11	5700	355	1451880.1999999995	Dudoso	Uso mensual	east Bohemia	8388	87	95907	59.1
...

682 filas

|

3,52 segundos de tiempo de ejecución

Actualizado hace 1 hora

Creamos un Dataframe Pyspark

```

1 #import SparkSession
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from pyspark.sql import SparkSession
5 from pyspark.sql.types import StructType, StructField, StringType, LongType, IntegerType, DoubleType, FloatType
6
7 # Create SparkSession
8 spark = SparkSession.builder \
9     .master("local[1]") \
10     .appName("SparkByExamples.com") \
11     .getOrCreate()
12 df=spark.sql("SELECT * FROM transcompleta WHERE Confiabilidad != 'No se presto';")

```

df: pyspark.sql.dataframe.DataFrame = [Cuenta: string, Nro_movimientos: long ... 10 campos adicionales]

Comando ejecutado en 0,29 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:50:35 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Verificamos que las extensiones sean correctas:

```
1 df.printSchema()
```

```
root
|-- Cuenta: string (nullable = true)
|-- Nro_movimientos: long (nullable = false)
|-- Total_Dinero_movido: double (nullable = true)
|-- Confiabilidad: string (nullable = false)
|-- Frecuencia: string (nullable = true)
|-- Region: string (nullable = true)
|-- Salario_Promedio: string (nullable = true)
|-- Empresarios_en_miles: string (nullable = true)
|-- Habitantes: string (nullable = true)
|-- Urbanizacion: string (nullable = true)
|-- Desempleo: string (nullable = true)
|-- Crimenes: string (nullable = true)
```

En este caso no lo son por lo que cambiamos el tipo de variable por las correctas:

```
df2= spark.sql("select Double(Cuenta),Double(Nro_Movimientos),Double(Total_Dinero_movido), String(Confiabilidad),String(Frecuencia), String(Region),
Double(Salario_Promedio),Double(Empresarios_en_miles),Double(Habitantes),Double(Urbanizacion),Double(Desempleo),Double(Crimenes) from transcompleta")
```

```
1 df2.printSchema()
```

```
root
|-- Cuenta: double (nullable = true)
|-- Nro_Movimientos: double (nullable = false)
|-- Total_Dinero_movido: double (nullable = true)
|-- Confiabilidad: string (nullable = false)
|-- Frecuencia: string (nullable = true)
|-- Region: string (nullable = true)
|-- Salario_Promedio: double (nullable = true)
|-- Empresarios_en_miles: double (nullable = true)
|-- Habitantes: double (nullable = true)
|-- Urbanizacion: double (nullable = true)
|-- Desempleo: double (nullable = true)
|-- Crimenes: double (nullable = true)
```

Realizamos el tratamiento de los valores nulos, como no se tenían la tabla quedo igual:

```
1 df2.count()
```

▸ (6) trabajos de Spark

Out[36]: 4500

Comando ejecutado en 2,60 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:50:35 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Cmd 15

```
1 df3 = df2.na.drop()
```

▸ df3: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 10 campos adicionales]

Comando ejecutado en 0,09 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:50:35 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Cmd 16

```
1 df2.count()
```

▸ (6) trabajos de Spark

Out[38]: 4500

Comando ejecutado en 2,62 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:50:35 en «Juan de Dios Delgadillo's Personal Compute Cluster»

5.2 Agrupamiento de variables Cualitativas

Agrupamos variables cualitativas

Agrupamiento de Variables Cualitativas

```
1 df3.groupBy(F.col('Cuenta')).count().show(5)
2 df3.groupBy(F.col('Confiabilidad')).count().show(5)
3 df3.groupBy(F.col('Frecuencia')).count().show(5)
4 df3.groupBy(F.col('Region')).count().show(5)
```

```
+-----+-----+
|Cuenta|count|
+-----+-----+
|2862.0|    1|
|1051.0|    1|
|1761.0|    1|
|3980.0|    1|
|4066.0|    1|
+-----+-----+
only showing top 5 rows
```

```
+-----+-----+
|Confiabilidad|count|
+-----+-----+
|Dudoso|    45|
|Inconfiable|    31|
|Confiable|   403|
|No se presto| 3818|
|Excelente candidato| 203|
+-----+-----+
```

```
+-----+-----+
|Frecuencia|count|
+-----+-----+
|Uso mensual| 4167|
|Uso semanal|  240|
|Frecuente|   93|
+-----+-----+
```

```
+-----+-----+
|Region|count|
+-----+-----+
|Prague|   554|
|north Bohemia| 457|
|east Bohemia|  544|
|south Moravia| 778|
|north Moravia| 793|
+-----+-----+
```

5.3 Agrupamiento de variables cuantitativas

Agrupamos variables cuantitativas y las describimos por el método describe (), para obtener sus valores estadísticos:

Agrupamiento de Variables Cuantitativas

Python ▶ ▼ ✕ ✖

```
1 df3.select(['Nro_Movimientos','Total_Dinero_movido','Salario_Promedio','Empresarios_en_miles','Habitantes','Urbanizacion','Desempleo','Crimenes']).describe().show(10)
```

▶ (6) trabajos de Spark

summary	Nro_Movimientos	Total_Dinero_movido	Salario_Promedio	Empresarios_en_miles	Habitantes	Urbanizacion	Desempleo	Crimenes
count	4500	4500	4500	4500	4500	4500	4500	4500
mean	234.73777777777778	1390620.7911777825	9515.204	121.11555555555556	269243.1973333333	69.23924444444448	3.4967200000000008	16307.450888888889
stddev	126.84944300580064	1328642.2803681789	1326.411186235079	23.15707792624199	358316.5233030266	19.77112038995027	2.126928354402439	31281.88848226953
min	9.0	29400.0	8110.0	81.0	42821.0	33.9	0.43	888.0
max	675.0	7619102.4	12541.0	167.0	1204953.0	100.0	9.4	99107.0

6 Modelado

El modelado de machine Learning desempeña un papel fundamental en la gestión y el éxito de una empresa en la era actual de la tecnología y la información. La importancia radica en su capacidad para convertir los datos en información procesable y conocimiento valioso. Al aplicar algoritmos de machine Learning a los datos de una

empresa, se pueden lograr varios beneficios significativos. Esto puede ayudar al Banco Berka a obtener conocimiento acerca de los potenciales buenos y malos clientes para realizar préstamos. Sin embargo, es importante notar que los algoritmos de machine Learning no deberían usarse para decisiones que condenen a personas, que por el hecho de que una persona viva en un determinado distrito y que el resto de personas que viven en ese distrito no paguen sus deudas afecte a que se le realice un préstamo por variables que esa persona no controla.

6.1 Aprendizaje Supervisado

El aprendizaje supervisado es una rama del machine Learning donde se entrena un modelo utilizando datos etiquetados, es decir, datos donde se conoce la relación entre las entradas y las salidas deseadas. El objetivo es que el modelo aprenda a hacer predicciones precisas sobre nuevas entradas basadas en la información proporcionada por los datos etiquetados. Es ampliamente utilizado en clasificación y regresión, permitiendo, por ejemplo, predecir categorías de objetos o valores numéricos.

En este caso nuestra variable objetivo será el estado de las deudas de un cliente basado en si pago sus deudas anteriormente sin problemas o no, o si tienen una deuda y como va el cliente con esa deuda, basados en la siguiente tabla:

	Cuenta	Nro_movimientos	Total_Dinero_movido	Confiabilidad	Frecuencia	Region	Salario_Promedio	Empresarios_en_miles	Habitantes	Urbanizacion	Desempleo	Crimenes
1	813	392	1922343.0999999992	Excelente candidato	Uso mensual	south Bohemia	8427	107	93931	56.9	1.54	1913
2	544	254	2191771.8999999994	Excelente candidato	Uso mensual	north Bohemia	9272	118	105058	81.0	3.22	4505
3	9869	465	3044394.3	Confiable	Uso mensual	central Bohemia	8754	137	107870	58.0	4.31	3868
4	2051	224	2893258.8	Dudoso	Uso semanal	south Moravia	9624	145	197099	74.7	2.31	4265
5	7819	591	1298770.8999999994	Confiable	Uso mensual	north Bohemia	8965	104	85852	59.8	8.23	2822
6	1843	462	2764436.8999999994	Excelente candidato	Uso mensual	central Bohemia	8754	137	107870	58.0	4.31	3868
7	7753	500	6346623.2000000005	Excelente candidato	Uso mensual	north Moravia	10673	100	323870	100.0	5.44	18347
8	6118	408	3032913.2000000007	Inconfiable	Uso semanal	Prague	12541	167	1204953	100.0	0.43	99107
9	2912	369	5610280.7	Confiable	Uso mensual	north Moravia	10673	100	323870	100.0	5.44	18347
10	6555	438	4243596.7	Excelente candidato	Uso semanal	north Bohemia	8965	104	85852	59.8	8.23	2822
11	5700	355	1451880.1999999995	Dudoso	Uso mensual	east Bohemia	8388	87	95907	59.1	2.94	1668

Donde la confiabilidad es nuestra variable Y o etiqueta para el aprendizaje supervisado, el cual nos dirá sin un cliente es un excelente candidato, un buen candidato, un mal candidato, un pésimo candidato.

Realizamos la transformación One Hot encoding para las variables cualitativas:

```
1 from pyspark.ml.feature import VectorAssembler, OneHotEncoder, StringIndexer
```

Comando ejecutado en 0,09 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:30:04 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Cmd 20

```
1 Region_indexer = StringIndexer(inputCol= 'Region', outputCol= 'RegionIndex')
2 Region_encoder = OneHotEncoder(inputCol = 'RegionIndex', outputCol= 'RegionVec')
3
4 Frecuencia_indexer = StringIndexer(inputCol= 'Frecuencia', outputCol= 'FrecuenciaIndex')
5 Frecuencia_encoder = OneHotEncoder(inputCol = 'FrecuenciaIndex', outputCol= 'FrecuenciaVec')
6
7 Confiabilidad_indexer = StringIndexer(inputCol= 'Confiabilidad', outputCol= 'ConfiabilidadIndex')
```

Comando ejecutado en 0,21 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:30:06 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Definimos los features para el modelo:

```
1 assembler = VectorAssembler(inputCols = ['FrecuenciaVec', 'RegionVec', 'Cuenta', 'Nro_Movimientos', 'Total_Dinero_movido', 'Salario_Promedio',
'Empresarios_en_miles', 'Habitantes', 'Urbanizacion', 'Desempleo', 'Crimenes'], outputCol= 'features')
```

Comando ejecutado en 0,13 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:30:17 en «Juan de Dios Delgadillo's Personal Compute Cluster»

6.1.1 Regresión Logística

Instanciamos la función para el modelado de Regresión Logística

```
1 from pyspark.ml.classification import LogisticRegression
2 from pyspark.ml import Pipeline
```

Comando ejecutado en 0,08 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:28:55 en «Juan de Dios Delgadillo's Personal Compute Cluster»

.md 23

```
1 log_reg_berka = LogisticRegression(featuresCol='features', labelCol='ConfiabilidadIndex')
```

Comando ejecutado en 0,29 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»



Creamos un pipeline para ejecutar todo lo anterior en Spark:

```
1 pipeline = Pipeline(stages= [
2     Region_indexer,
3     Region_encoder,
4     Frecuencia_indexer,
5     Frecuencia_encoder,
6     Confiabilidad_indexer,
7     assembler,
8     log_reg_berka])
```

Comando ejecutado en 0,09 segundos -- por paperoski@hotmail.com el 16/9/2023,

Dividimos el dataset en training y testing:

```
1 train_data, test_data = df3.randomSplit([0.7,0.3])
```

- ▶  train_data: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 10 campos adicionales]
- ▶  test_data: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 10 campos adicionales]

Comando ejecutado en 0,09 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Entrenamos el modelo:

```
1 fit_model = pipeline.fit(train_data)
```

▶ (45) trabajos de Spark

▼ (1) ejecución de MLflow

Se ha registrado 1 [ejecución](#) de un [experimento](#) en MLflow. [Más información](#)

 1

Comando ejecutado en 9,76 minutos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Mostramos el resultado del entrenamiento:

```
1 results = fit_model.transform(test_data)
2 results.show(5)
```

► (5) trabajos de Spark

► results: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 19 campos adicionales]

[Cuenta Nro_Movimientos Total_Dinero_movido Confiabilidad Frecuencia Region Salario_Promedio Empresarios_en_miles Habitantes Urbanizacion Desempleo Crimenes RegionIndex RegionVec FrecuenciaIndex FrecuenciaVec ConfiabilidadIndex features rawPrediction]																		
probability prediction]																		
2.0 478.0 3151479.3 Excelente candidato Uso mensual Prague 12541.0 167.0 1204953.0 100.0 0.43 99107.0 4.0 (7,[4],[1.0]) 0.0 (2,[0],[1.0]) 2.0 [1.0,0.0,0.0,0.0,... [4.0687932553461																		
3... [0.95581567888692... 0.0																		
3.0 117.0 295021.800000000005 No se presto Uso mensual central Bohemia 9307.0 118.0 95616.0 51.4 4.43 3040.0 3.0 (7,[3],[1.0]) 0.0 (2,[0],[1.0]) 0.0 [1.0,0.0,0.0,0.0,... [5.0869440830965																		
1... [0.98392624812173... 0.0																		
7.0 130.0 1192039.9 No se presto Uso mensual south Moravia 8441.0 115.0 110643.0 51.9 4.48 2252.0 1.0 (7,[1],[1.0]) 0.0 (2,[0],[1.0]) 0.0 [1.0,0.0,0.0,1.0,... [4.8960348305958																		
8... [0.98085446591729... 0.0																		
8.0 254.0 1712906.0999999996 No se presto Uso mensual south Moravia 8720.0 116.0 161954.0 48.0 4.5 3651.0 1.0 (7,[1],[1.0]) 0.0 (2,[0],[1.0]) 0.0 [1.0,0.0,0.0,1.0,... [4.4475185800321																		
1... [0.97450341868074... 0.0																		

Comando ejecutado en 4,29 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Mostramos la predicción y el valor real para compararlo y evaluar el rendimiento del modelo:

```
1 me_eval = BinaryClassificationEvaluator(rawPredictionCol= 'prediction', labelCol = 'ConfiabilidadIndex')
```

Comando ejecutado en 0,09 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Cmd 30

```
1 results.select('ConfiabilidadIndex', 'prediction').show(10)
```

► (5) trabajos de Spark

[ConfiabilidadIndex prediction]		
2.0 0.0		
0.0 0.0		
0.0 0.0		
0.0 0.0		
0.0 0.0		
0.0 0.0		
0.0 0.0		
0.0 0.0		
2.0 0.0		
0.0 0.0		

only showing top 10 rows

Comando ejecutado en 3,50 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Evaluamos nuestro modelo:

Evaluacion del Modelo

Py

```
1 auc = me_eval.evaluate(results)
2 print("AUC:",auc)
```

► (12) trabajos de Spark

AUC: 0.6988409908357415

Comando ejecutado en 8,45 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

6.1.2 Random Forest

Instanciamos la función para el modelado de Regresión Logística y creamos un pipeline para ejecutar todo lo anterior en Spark:

```
1 from pyspark.ml.classification import RandomForestClassifier
2 rf = RandomForestClassifier(numTrees=10, maxDepth=6, labelCol="ConfiabilidadIndex", seed=42, leafCol="leafId")
3 pipeline = Pipeline(stages= [
4     Region_indexer,
5     Region_encoder,
6     Frecuencia_indexer,
7     Frecuencia_encoder,
8     Confiabilidad_indexer,
9     assembler,
10    rf])
```

Entrenamos el modelo aprovechando los features de regresión logística, ya que son los mismos:

```
1 fit_modelrf = pipeline.fit(train_data)
```

▶ (43) trabajos de Spark

▼ (1) ejecución de MLflow

Se ha registrado 1 ejecución de un experimento en MLflow. [Más información](#)

Comando ejecutado en 2,57 minutos -- por paperoski@hotmail.com el 16/9/2023, 10:54:37 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Mostramos el resultado del entrenamiento:

```
1 resultsrf = fit_modelrf.transform(test_data)
2 resultsrf.show(5)
```

▶ (5) trabajos de Spark

▶ resultsrf: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 20 campos adicionales]

Cuenta	Nro_Movimientos	Total_Dinero_movido	Confiabilidad	Frecuencia	Region	Salario_Promedio	Empresarios_en_miles	Habitantes	Urbanizacion	Desempleo	Crimenes	RegionIndex	RegionVec	FrecuenciaIndex	FrecuenciaVec	ConfiabilidadIndex	features	rawPredict
ion	probability	prediction	leafId															
2.0	478.0	3151479.3	Excelente candidato	Uso mensual	Prague	12541.0	167.0	1204953.0										
100.0	0.43	99107.0	4.0 (7,[4],[1.0])	0.0 (2,[0],[1.0])		2.0 [1.0,0.0,0.0,0.0,...	[7.0989400619878											
8...	[0.70989400619878...		0.0 [30.0,32.0,18.0,2...															
3.0	117.0	295021.80000000005	No se presto	Uso mensual	central Bohemia	9307.0	118.0	95616.0										
51.4	4.43	3040.0	3.0 (7,[3],[1.0])	0.0 (2,[0],[1.0])		0.0 [1.0,0.0,0.0,0.0,...	[9.6462233081163											
1...	[0.96462233081163...		0.0 [3.0,0.0,0.0,3.0,...															
7.0	130.0	1192039.9	No se presto	Uso mensual	south Moravia	8441.0	115.0	110643.0										
51.9	4.48	2252.0	1.0 (7,[1],[1.0])	0.0 (2,[0],[1.0])		0.0 [1.0,0.0,0.0,1.0,...	[9.3619467942391											
7...	[0.93619467942391...		0.0 [14.0,0.0,3.0,3.0,...															
8.0	254.0	1712906.0999999996	No se presto	Uso mensual	south Moravia	8720.0	116.0	161954.0										
48.0	4.5	3651.0	1.0 (7,[1],[1.0])	0.0 (2,[0],[1.0])		0.0 [1.0,0.0,0.0,1.0,...	[9.0686225188838											
4...	[0.90686225188838...		0.0 [15.0,0.0,3.0,17....															

Evaluamos el modelo:

Evaluación del Modelo (Random Forest)

```
1 auc = me_eval.evaluate(resultsrf)
2 print("AUC:", auc)
```

► (12) trabajos de Spark

AUC: 0.7754314801558895

Comando ejecutado en 8,75 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:37 en «Juan de Dios Delgadillo's Personal Compute Cluster»

6.1.3 Reflexión comparativa de modelos

La elección entre Random Forest y Regresión Logística depende en última instancia de la naturaleza del problema y los datos disponibles. Random Forest, al ser un conjunto de árboles de decisión, puede manejar relaciones no lineales y complejas en los datos, lo que lo hace valioso en problemas de clasificación y regresión donde las interacciones entre variables son difíciles de modelar linealmente. Sin embargo, la Regresión Logística es más simple, interpretable y eficiente en problemas con relaciones lineales o cuando se necesita una comprensión clara de las relaciones entre variables.

Pudimos observar que el modelo de Random forest tuvo un mejor rendimiento que el de regresión Logística, esto es debido a que el primero es un método de ensamble, obteniendo **AUC de 0.78 y 0.69** respectivamente.

6.2 Machine Learning no supervisado

El aprendizaje no supervisado es una rama del machine learning en la que se utilizan datos no etiquetados para descubrir patrones y estructuras ocultas. A diferencia del aprendizaje supervisado, no hay una salida objetivo conocida, lo que significa que el modelo busca agrupar datos similares o reducir la dimensionalidad para revelar relaciones intrínsecas entre los datos.

En este caso utilizamos un tipo de dato no analizado en la sección anterior. Los datos de aprendizaje supervisado solo estaban basados en las personas que ya habían realizado antes un préstamo, pero más del 50% de los clientes nunca realizaron un préstamo y no tiene historial del mismo para saber cuál es un posible candidato, para ese tipo de clientes realizamos un algoritmo de segmentación que nos clasifica los datos en 4 grupos de donde nosotros podríamos inferir cuál es un mejor cliente.

6.2.1 Segmentación

Segmentación

```
1 from pyspark.ml.clustering import KMeans
2 from pyspark.ml.evaluation import ClusteringEvaluator
```

Comando ejecutado en 0,09 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:28:25 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Volvemos a importar los datos, pero esta vez seleccionan solo aquellos que nunca realizaron la solicitud de préstamo:

```
Python ▶ ▼ - ✕
1 dfclus= spark.sql("select Double(Cuenta),Double(Nro_Movimientos),Double(Total_Dinero_movido), String(Confiabilidad),String
  (Frecuencia), String(Region), Double(Salario_Promedio),Double(Empresarios_en_miles),Double(Habitantes),Double(Urbanizacion),Double
  (Desempleo),Double(Crimenes) from transcompleta WHERE Confiabilidad = 'No se presto';")
2 kmeans = KMeans().setK(5).setSeed(1)

▶ dfclus: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 10 campos adicionales]
Comando ejecutado en 0,43 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:28:29 en «Juan de Dios Delgadillo's Personal Compute Cluster»
```

Dividimos los datos en training y testing:

```
1 train_data, test_data = dfclus.randomSplit([0.7,0.3])

▶ train_data: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 10 campos adicionales]
▶ test_data: pyspark.sql.dataframe.DataFrame = [Cuenta: double, Nro_Movimientos: double ... 10 campos adicionales]
Comando ejecutado en 0,15 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:28:40 en «Juan de Dios Delgadillo's Personal Compute Cluster»
```

Realizamos un pipeline para ejecutar el modelo, aprovechamos los features de regresión logística pero retiramos la variable etiqueta:

```
1 pipeline2 = Pipeline(stages= [
2     Region_indexer,
3     Region_encoder,
4     Frecuencia_indexer,
5     Frecuencia_encoder,
6     assembler,
7     kmeans])

Comando ejecutado en 0,07 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:30:24 en «Juan de Dios Delgadillo's Personal Compute Cluster»
```

Entrenamos el modelo

```
1 fit_model2 = pipeline2.fit(train_data)

▶ (46) trabajos de Spark

▼ (1) ejecución de MLflow
  Se ha registrado 1 ejecución de un experimento en MLflow. Más información

🔔 1
Comando ejecutado en 4,31 minutos -- por paperoski@hotmail.com el 16/9/2023, 11:30:28 en «Juan de Dios Delgadillo's Personal Compute Cluster»
```

Mostramos los resultados de la predicción:

```
1 predictions2.show(10)

▶ (5) trabajos de Spark
```

Cuenta	Nro_Movimientos	Total_Dinero_movido	Confiabilidad	Frecuencia	Region	Salario_Promedio	Empresarios_en_miles	Habitantes	Urbanizaci
on Desempleo Crimenes RegionIndex	RegionVec FrecuenciaIndex FrecuenciaVec ConfiabilidadIndex	features prediction							
5.0	84.0	166881.40000000002	No se presto	Uso mensual south Bohemia	9045.0	124.0	58796.0	5	
1.9	3.6	1879.0	7.0 (7,[,])	0.0 (2,[0],[1.0])	0.0 (18,[0,9,10,11,12...	1			
6.0	246.0	647567.1000000002	No se presto	Uso mensual east Bohemia	8541.0	131.0	121947.0	7	
0.5	2.97	3839.0	4.0 (7,[4],[1.0])	0.0 (2,[0],[1.0])	0.0 [1.0,0.0,0.0,0.0,...	1			
7.0	130.0	1192039.9	No se presto	Uso mensual south Moravia	8441.0	115.0	110643.0	5	
1.9	4.48	2252.0	1.0 (7,[1],[1.0])	0.0 (2,[0],[1.0])	0.0 [1.0,0.0,0.0,1.0,...	4			
10.0	156.0	1040605.1	No se presto	Uso mensual south Moravia	9897.0	140.0	387570.0	10	
0.0	1.96	18696.0	1.0 (7,[1],[1.0])	0.0 (2,[0],[1.0])	0.0 [1.0,0.0,0.0,1.0,...	4			
11.0	186.0	258100.09999999998	No se presto	Uso mensual north Moravia	8369.0	107.0	127369.0	5	
1.2	5.88	2807.0	0.0 (7,[0],[1.0])	0.0 (2,[0],[1.0])	0.0 [1.0,0.0,1.0,0.0,...	1			
15.0	365.0	2329572.8999999994	No se presto	Uso mensual north Moravia	10177.0	81.0	285387.0	8	
9.9	7.75	10108.0	0.0 (7,[0],[1.0])	0.0 (2,[0],[1.0])	0.0 [1.0,0.0,1.0,0.0,...	4			

Finalmente evaluamos el modelo con el coeficiente Silhouette:

Evaluacion del Modelo (KMeans)

Python ▶ ▼ - ✕

```
1 evaluador = ClusteringEvaluator()
```

Comando ejecutado en 0,17 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:35:59 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Cmd 44

```
1 silhouette = evaluador.evaluate(predictions2)
2 print("El coeficiente Silhouette usando distancias euclidianas al cuadrado es = " + str(silhouette))
```

▶ (16) trabajos de Spark

El coeficiente Silhouette usando distancias euclidianas al cuadrado es = 0.7639184006399297

Comando ejecutado en 13,57 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:35:59 en «Juan de Dios Delgadillo's Personal Compute Cluster»

7 Evaluación de los modelos

Como ya se había realizado este procedimiento en la sección anterior, en la presente sección se mostraran los resultados de la evaluación de cada modelo:

Mediante aprendizaje supervisado a través de Regresión Logística:

Evaluacion del Modelo

Py

```
1 auc = me_eval.evaluate(results)
2 print("AUC:",auc)
```

▶ (12) trabajos de Spark

AUC: 0.6988409908357415

Comando ejecutado en 8,45 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:36 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Mediante aprendizaje supervisado a través de Random Forest:

Evaluacion del Modelo (Random Forest)

```
1 auc = me_eval.evaluate(resultsrfr)
2 print("AUC:",auc)
```

▶ (12) trabajos de Spark

AUC: 0.7754314801558895

Comando ejecutado en 8,75 segundos -- por paperoski@hotmail.com el 16/9/2023, 10:54:37 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Mediante aprendizaje no supervisado a través de clustering con Kmeans:

Evaluacion del Modelo (KMeans)

Python ▶ ▼ - ✕

```
1 evaluador = ClusteringEvaluator()
```

Comando ejecutado en 0,17 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:35:59 en «Juan de Dios Delgadillo's Personal Compute Cluster»

Cmd 44

```
1 silhouette = evaluador.evaluate(predictions2)
2 print("El coeficiente Silhouette usando distancias euclidianas al cuadrado es = " + str(silhouette))
```

▶ (16) trabajos de Spark

El coeficiente Silhouette usando distancias euclidianas al cuadrado es = 0.7639184006399297

Comando ejecutado en 13,57 segundos -- por paperoski@hotmail.com el 16/9/2023, 11:35:59 en «Juan de Dios Delgadillo's Personal Compute Cluster»

8 Conclusiones

Aprovechando la tabla minable obtenida en el módulo I, se dividió dicha tabla en 2 partes, una para aquellos que ya tenían un historial de préstamo bancario y el otro para aquellos que no. Se utilizó aprendizaje supervisado para la primera parte de la tabla (aquellos con historial de préstamos) y aprendizaje no supervisado para aquellos que nunca pidieron un préstamo al banco. En aprendizaje supervisado se utilizó 2 algoritmos de machine Learning: Random Forest y Regresión Logística y en la comparación entre los mismos se concluye que Random Forest superó significativamente a la Regresión Logística en términos de rendimiento predictivo, demostrando una mayor capacidad para capturar relaciones complejas en los datos. En el aprendizaje no supervisado se utilizó el algoritmo de Kmeans el cual tuvo un rendimiento satisfactorio para clasificar a los clientes que nunca realizaron una solicitud de préstamo.

9 Recomendaciones

Se recomienda hacer una segunda evaluación a aquellas personas a las que el algoritmo clasifique como malos candidatos, debido a la precisión del algoritmo que a pesar de ser cerca al 80% no es 100% confiable, además que un algoritmo de aprendizaje automático no debe decidir acerca del futuro de las personas, y aquellas que se vean como malos candidatos deben tener la oportunidad de apelar y que se les diga la razón de porque no se puede realizar el préstamo.