PROYECTO FINAL DATA SCIENCE

CODER HOUSE

Alumnos: Alcántara, Luciano
Rabazzi, Juan Ignacio

Comisión: 29820

Contenido

1 OBJETIVO COMERCIAL

Identificar el riesgo crediticio de un cliente.

2 CONTEXTO COMERCIAL

Para entregar una tarjeta de crédito o un préstamo los bancos deben evaluar la situación crediticia de los clientes, para lo cual solicitan el servicio de una consultora encargada de evaluar el riesgo crediticio.

Las evaluaciones de riesgos crediticios son un método común de control de riesgos en la industria financiera. La información personal y los datos presentados por los solicitantes de tarjetas de crédito es utilizada para predecir la probabilidad de futuros incumplimientos y préstamos de tarjetas de crédito. A partir de esta información, el banco puede decidir si emite una tarjeta de crédito al solicitante, ya que los puntajes de crédito pueden cuantificar objetivamente la magnitud del riesgo.

En términos generales, las evaluaciones de riesgos crediticios se basan en datos históricos. Un método para clasificación crediticia es el modelo Logístico, ya que es adecuado para tareas de clasificación binaria y puede calcular los coeficientes de cada característica.

Por otro lado, en la actualidad, con el desarrollo de algoritmos de machine learning se han introducido métodos predictivos como Boosting, Random Forest y Support Vector Machines en la evaluación de riesgo crediticio, aunque en ocasiones estos métodos no tienen una buena transparencia y pueden ser difícil proporcionar a los clientes y bancos una razón para el rechazo o la aceptación de la emisión de una tarjeta de crédito.

Finalmente, el objetivo de este proyecto es construir un modelo de Machine Learning para predecir si los clientes solicitantes de tarjetas de crédito deben ser aceptados o rechazados, para lo cual se deberá realizar una limpieza de la base de datos y luego definir el modelo más adecuado.

3 PROBLEMÁTICA COMERCIAL

3.1 PREGUNTAS HIPÓTESIS GENERALES

- ¿Qué se quiere predecir con el uso de la base de datos?

- ¿Cuál es el modelo más adecuado para tratar los datos?
- ¿Qué características tienen los clientes que son rechazados para obtener un crédito?

3.2 PREGUNTAS HIPÓTESIS DEL NEGOCIO

- ¿Cuál es la proporción de clientes aceptados y rechazados?
- ¿Cuál es la diferencia de ingresos según el género?
- ¿Existe relación entre la cantidad de hijos y los ingresos de los clientes?
- ¿Cuál es la diferencia de ingresos según el género y el estado de sus deudas (status)?
- ¿Es correcto afirmar que a mayor nivel de educación los clientes existen menos rechazos de clientes?
- ¿En qué tipo de vivienda viven los clientes?
- ¿Qué tipo de trabajo tienen los clientes?
- ¿Qué tipo de estado civil tienen los clientes?
- ¿Existe algún patrón en los ingresos de los clientes rechazados?

4 CONTEXTO ANALÍTICO

4.1 URL BASE DE DATOS:

https://www.kaggle.com/code/reemmuhammad2/creditcardapprovalprediction/data

4.2 DESCRIPCIÓN DE VARIABLES

Tabla: aplication_record

- ID: número de cliente.
- CODE_GENDER: género del cliente (M o F).
- FLAG_OWN_CAR: indica si el cliente posee un auto.
- FLAG_OWN_ REALTY: indica si el cliente posee una propiedad.
- CNT_CHILDREN: cantidad de hijos.
- AMT_INCOME_TOTAL: ingreso anual del cliente.
- NAME_INCOME_TYPE: Categoría de ingresos.

- NAME_EDUCATION_TYPE: tipo de educación del cliente.
- NAME_FAMILY_STATUS: estado civil del cliente.
- NAME_HOUSING_TYPE: indica si el cliente es inquilino.
- DAYS_BIRTH: muestra el día del cumpleaños del cliente. Está expresado como una cuenta atrás desde el día actual (-1 significa ayer).
- DAYS_EMPLOYED: días que el cliente lleva trabajando. Está expresado como una cuenta atrás desde el día actual (-1 significa ayer). Si es positivo significa que está desempleado.
- FLAG_MOBIL: indica si el cliente posee celular.
- FLAG_WORK_PHONE: indica si el celular del cliente es de trabajo.
- FLAG_PHONE: indica si el cliente tiene teléfono fijo.
- FLAG_EMAIL: indica si el cliente tiene email.
- OCCUPATION_TYPE: ocupación del cliente.
- CNT_FAM_MEMBERS: cantidad de miembros en la familia del cliente.

Tabla: credit_record

- ID: número de cliente.
- MONTHS_BALANCE: mes correspondiente al STATUS. El mes de los datos extraídos es el punto de partida (0 es el mes actual, -1 es el mes anterior).
- STATUS: indica el estado de la deuda del cliente cuando la canceló. 0: 1-29 días de atraso 1: 30-59 días de atraso 2: 60-89 días de atraso 3: 90-119 días de atraso 4: 120-149 días de atraso 5: Deudas atrasadas o incobrables, canceladas por más de 150 días C: pagado ese mes X: No hay préstamo para el mes.

4.3 SCRIPT EN PYTHON

Se adjunta un archivo. ipynb con el script del proyecto.

4.4 LECTURA DEL DATASET

1- Se importan las librerías necesarias para trabajar en el dataset
2- Se cargan los archivos .csv “aplication_record” y “credir_record”
3- Lectura de los datasets. Identifico las columnas del dataset aplication_record y las columnas del dataset credit_record.

5 ANÁLISIS EXPLORATORIO DE DATOS (EDA)

5.1 LIMPIEZA DE DATOS APLICATION

1- Se corrobora si existen IDs duplicados.
2- Se eliminan los IDs duplicados.
3- Se identifican la cantidad de datos nulos.
4- Se crean 2 dataframes auxiliares en donde se rellenan los valores nulos.
5- Se crea el dataframe aplication, concatenando los auxiliares.
6- Se eliminan los valores nulos y duplicados por ID.
7- Se eliminan los clientes repetidos.
8- Se transforman los nombres de las columnas a minusculas, para que sea más fácil trabajar.
9- Se crea la columna data_day, en donde se indica el día en el cual fueron obtenidos los datos.
10- Conversión de la columna data_day a dato de tiempo.
11- Se crea la columna birthday mediante un Bucle.
12- Se determina la cadinalidad de los datos.
13- Se convierten los datos al tipo correcto.

5.2 LIMPIEZA DE DATOS CREDIT

1- Se corrobora si existen registros duplicados.
2- Se eliminan los registros duplicados.
3- Se identifican la cantidad de datos nulos.
4- Se transforman los nombres de las columnas a minusculas, para que sea más fácil trabajar.

5- Se crea la columna data_day, en donde se indica el día en el cual fueron obtenidos los datos.
6- Conversión de la columna data_day a dato de tiempo.
7- Se crea la columna month_balance_date mediante un Bucle.
8- Se determina la cadinalidad de los datos.
9- Se convierten los datos al tipo correcto.

5.3 UNIÓN DE LOS DATASET Y CREACIÓN DE COLUMNAS

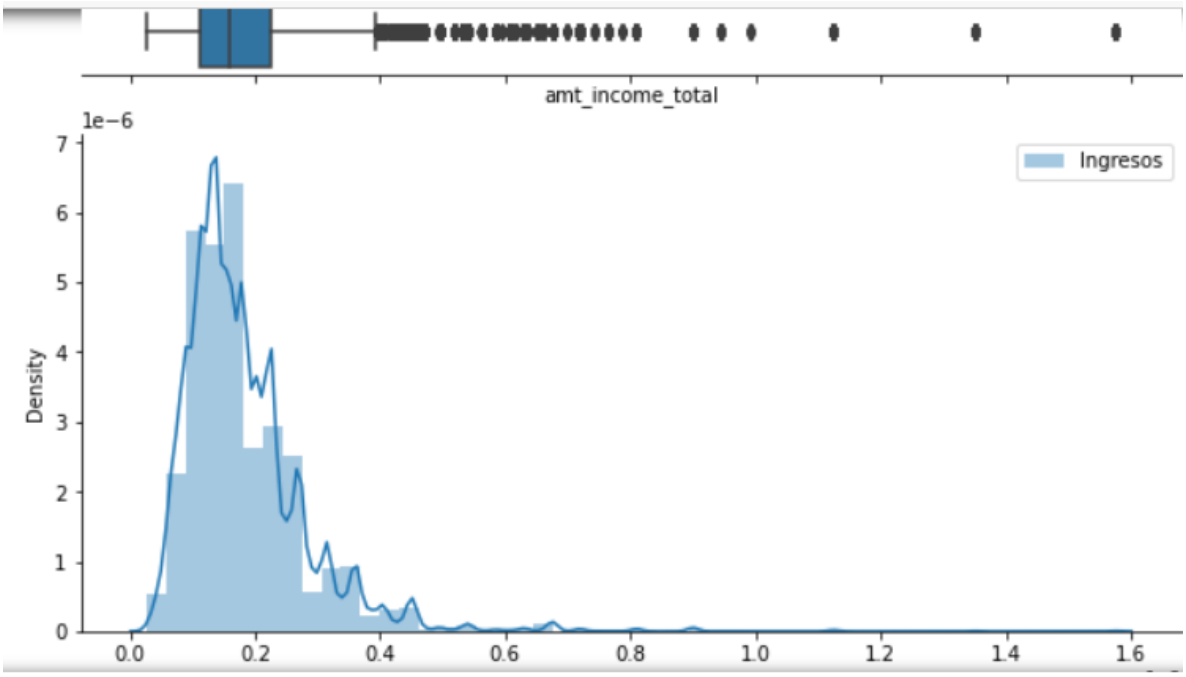
1- Se realizó el merge para los dataframes aplication y credit, a partir del campo id.
2- Se calculó la edad de los clientes (restando las columnas Data_day - Birthday).
3- Se definieron categorías para los ingresos de los clientes.

5.4 IDENTIFICACIÓN DE DATOS OUTLIERS

Se procedió a realizar la identificación de datos outliers de las siguientes columnas del dataset:

- amt_income_total
- cnt_children
- cnt_fam_members

Primero se utiliza el método gráfico para visualizar los outilers del dataframe en donde se obtuvo:



Finalmente, para identificar los outiliers dentro del dataframe se calculó el umbral superior e inferior de cada atributo y luego se utilizó el método intercuartílico.

6 GRÁFICOS

1- Se obtienen los registros únicos de los clientes.

2- Se obtiene las estadísticas básicas del dataframe, en donde se muestra:

- Para todas las variables la cantidad de datos.
- Para las variables categóricas se muestra la cantidad de valores único, el valor que se repite con mayor frecuencia y su frecuencia.
- Para las variables continuas se muestra le media, el desvío estándar, los valores máximos, valores mínimos y cuartiles.

	cnt_children	amt_income_total	days_birth	days_employed	flag_mobil	flag_work_phone	flag_phone	flag_email	cnt_fam_members	edad_dias
count	9709.000000	9.709000e+03	9709.000000	9709.000000	9709	9709	9709	9709	9709.000000	9709.000000
unique	NaN	NaN	NaN	NaN	1	2	2	2	NaN	NaN
top	NaN	NaN	NaN	NaN	True	False	False	False	NaN	NaN
freq	NaN	NaN	NaN	NaN	9709	7598	6916	8859	NaN	NaN
mean	0.422804	1.812282e+05	-15991.811618	61732.846328	NaN	NaN	NaN	NaN	2.182614	15991.811618
std	0.767019	9.927731e+04	4246.224468	139656.322958	NaN	NaN	NaN	NaN	0.932918	4246.224468
min	0.000000	2.700000e+04	-25152.000000	-15713.000000	NaN	NaN	NaN	NaN	1.000000	7489.000000
25%	0.000000	1.125000e+05	-19565.000000	-2995.000000	NaN	NaN	NaN	NaN	2.000000	12440.000000
50%	0.000000	1.575000e+05	-15611.000000	-1374.000000	NaN	NaN	NaN	NaN	2.000000	15611.000000
75%	1.000000	2.250000e+05	-12440.000000	-339.000000	NaN	NaN	NaN	NaN	3.000000	19565.000000
max	19.000000	1.575000e+06	-7489.000000	365243.000000	NaN	NaN	NaN	NaN	20.000000	25152.000000

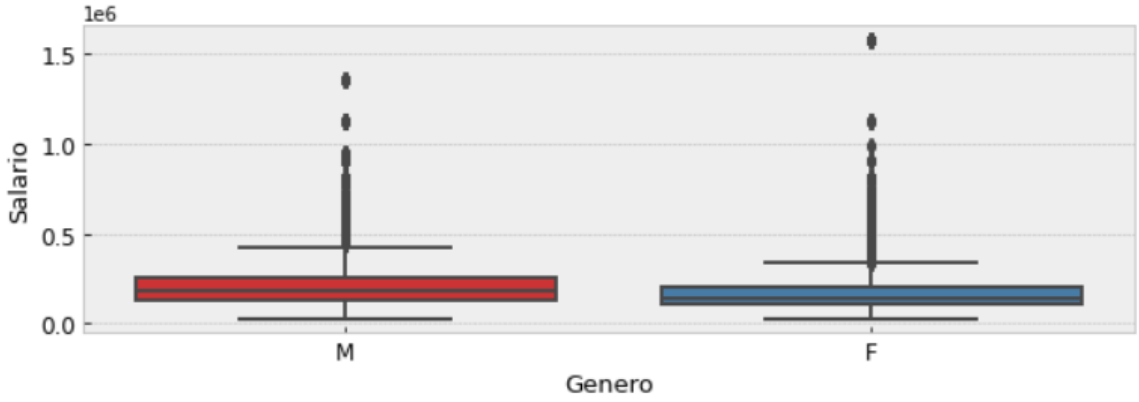
	edad	amt_income_total_outliers	cnt_children_outliers	cnt_fam_members_outliers
count	9709.000000	9709.000000	9709.000000	9709.000000
unique	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN
mean	43.309198	0.032753	0.015656	0.014626
std	11.635746	0.177999	0.124145	0.120055
min	20.000000	0.000000	0.000000	0.000000
25%	34.000000	0.000000	0.000000	0.000000
50%	42.000000	0.000000	0.000000	0.000000
75%	53.000000	0.000000	0.000000	0.000000
max	68.000000	1.000000	1.000000	1.000000

- Se declara el estilo de los gráficos y luego se procede a realizarlos.

6.1 COMPARACIÓN DE INGRESOS DE LOS CLIENTES ENTRE HOMBRES Y MUJERES

Se realizó un boxplot con seaborn que grafica los ingresos anuales entre hombres y mujeres.

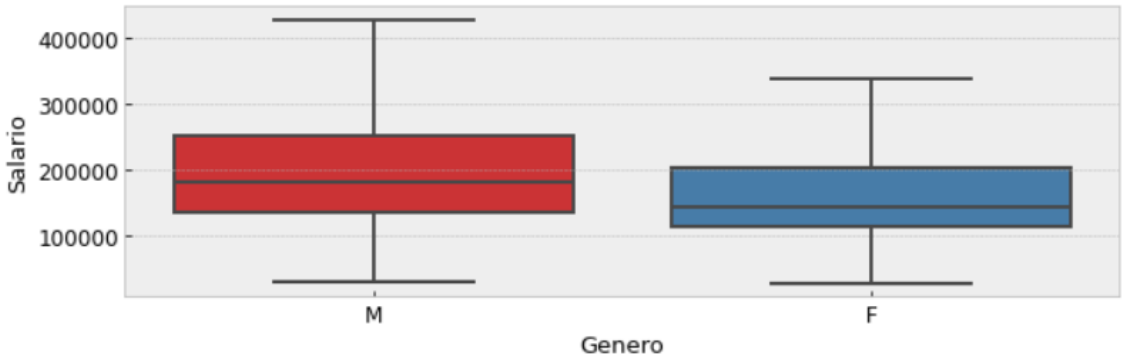
6.1.1 Boxplot con todos los datos



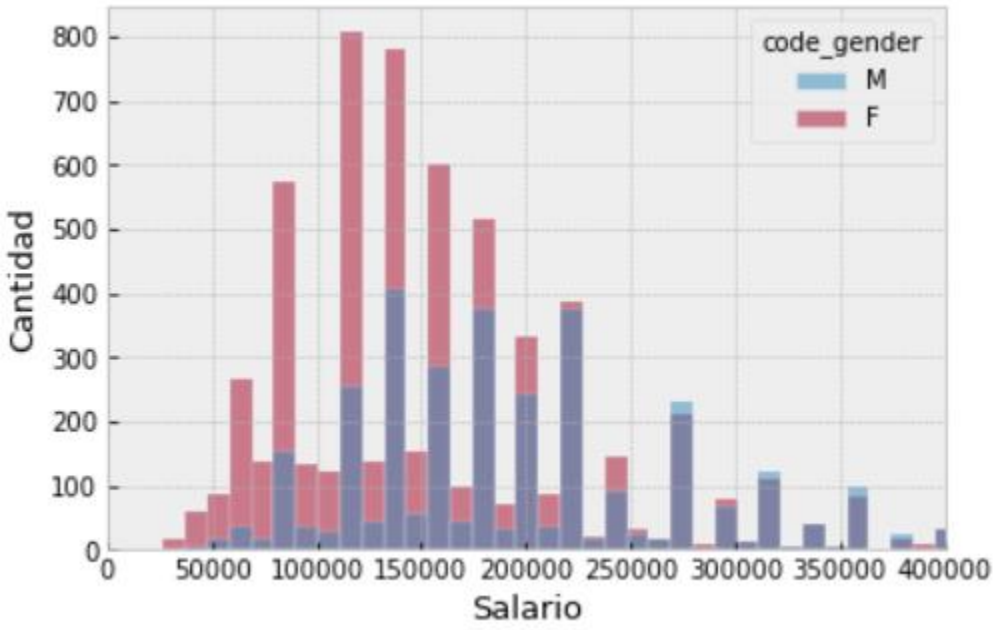
6.1.2 Boxplot sin datos outliers

Como el dataframe posee valores atípicos en el campo amt_income_total, el boxplot obtenido no permite visualizar correctamente las diferencias entre los

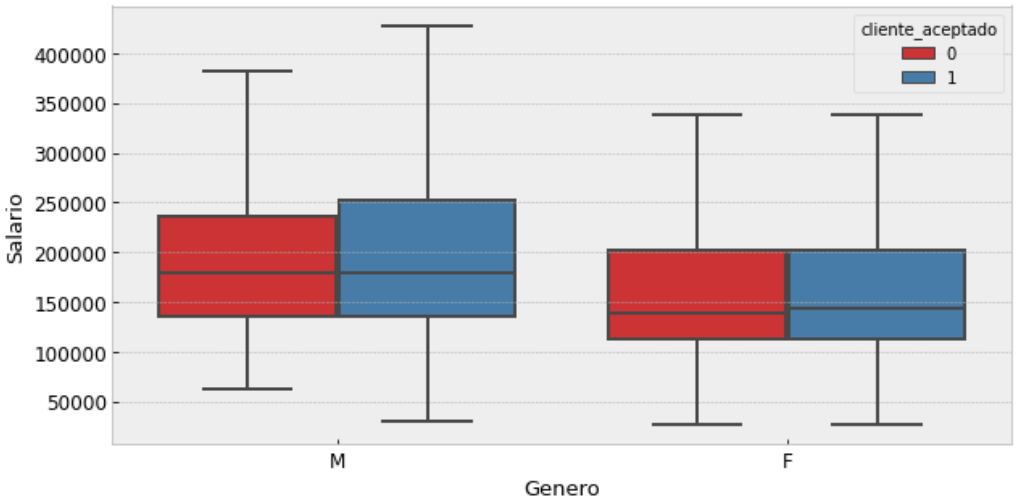
salarios de hombres y mujeres, por este motivo se realiza el siguiente boxplot sin outliers:



Por otro lado, se realizó un histográma con seaborn en donde se muestran los ingresos anuales entre hombres y mujeres menores a 400.000 dólares, debido a que los ingresos superiores son considerados outliers.



6.1.3 Boxplot clientes aceptados/rechazados por género



Referencia: 0, cliente rechazado. 1, cliente aceptado.

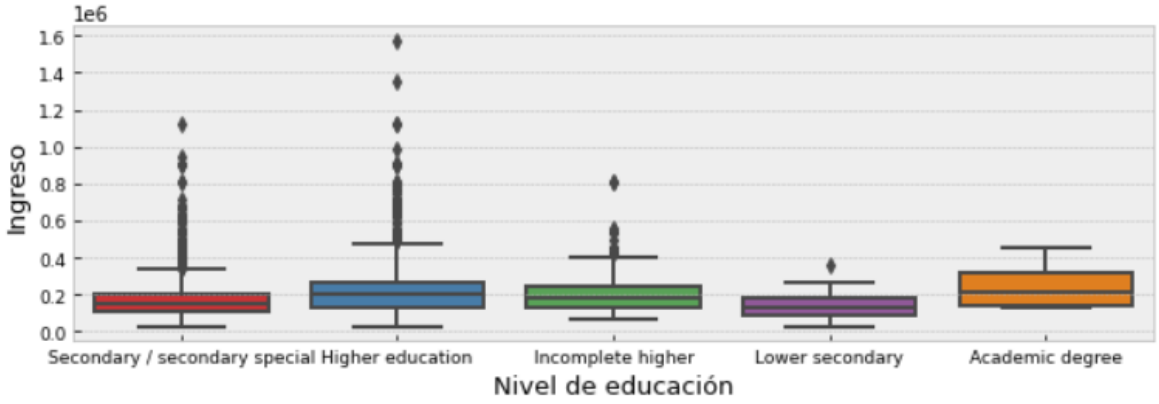
6.1.4 Conclusiones

- La media de ingresos anuales del género masculino es superior a la media de ingresos anules del género femenino.
- En el boxplot con datos outliers se puede observar que el mayor ingreso anual corresponde al género femenino.

6.2 INGRESOS DE LOS CLIENTES DIFERENCIADO POR NIVEL DE EDUCACIÓN

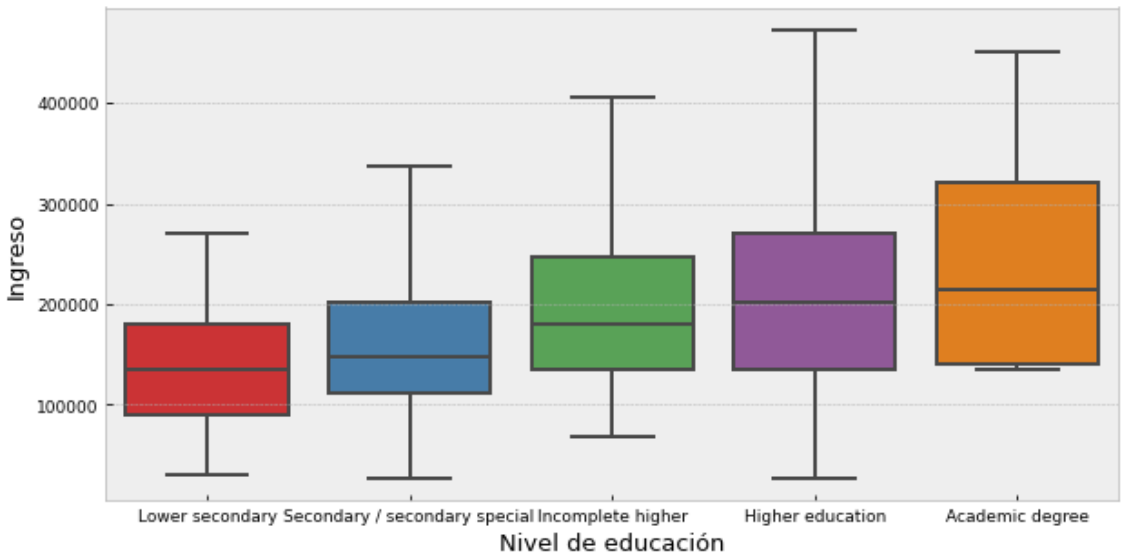
Se realizó un boxplot con seaborn que grafica los ingresos anuales diferenciado por nivel de educación de las personas.

6.2.1 Boxplot con todos los datos



6.2.2 Boxplot sin datos outliers

Como el dataframe posee valores atípicos en el campo amt_income_total, el boxplot obtenido no permite visualizar correctamente las diferencias entre los salarios según el nivel de educación, por este motivo se realiza el siguiente boxplot sin outliers:



6.2.3 Conclusiones

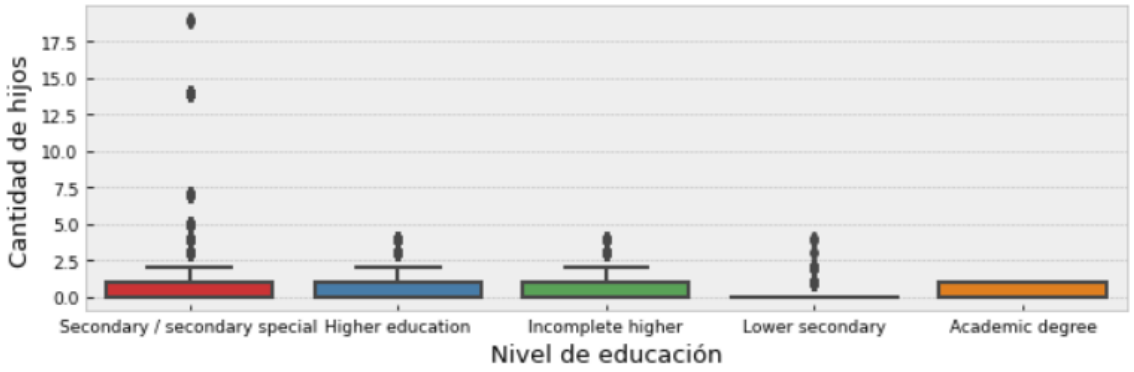
- Como se puede observar, a medida que aumenta el nivel de educación, mayores son los ingresos de los clientes.

- Por otro lado, se puede observar que el mayor ingreso anual corresponde a una persona que tiene secundario como nivel de educación.

6.3 CANTIDAD DE HIJOS DIFERENCIADO POR NIVEL DE EDUCACIÓN DE LOS CLIENTES

Se realizó un boxplot con seaborn, el cual grafica la cantidad de hijos diferenciado por nivel de educación de las personas.

6.3.1 Boxplot con todos los datos

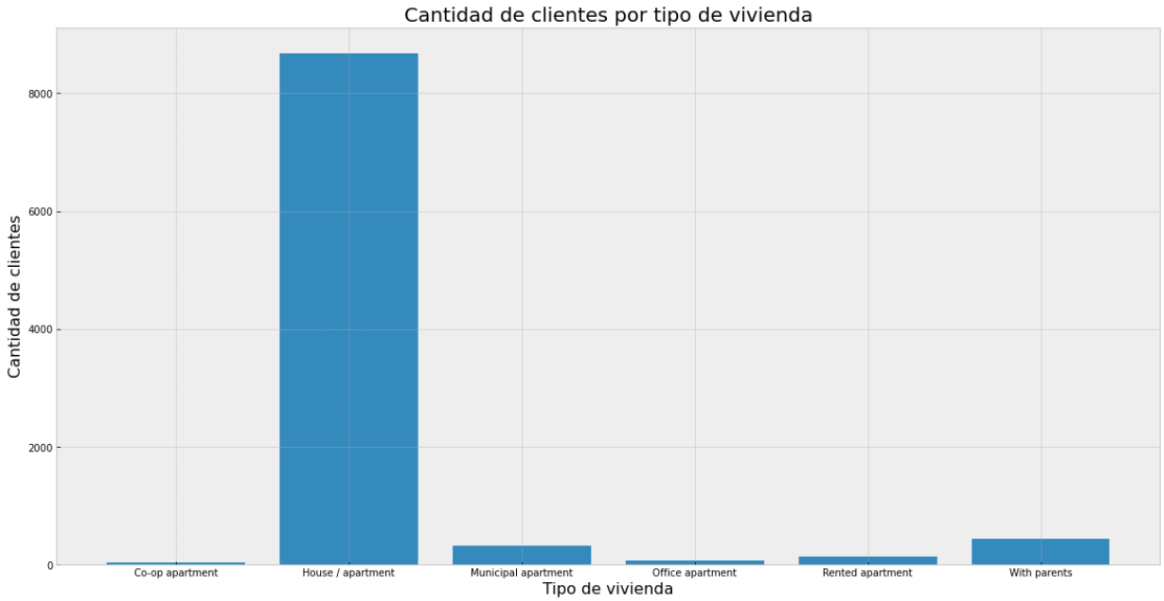


6.3.2 Conclusiones

- Se puede observar que las personas con un nivel de educación secundario son las que mayor cantidad de datos outliers tienen.
- No se puede concluir que la cantidad de hijos de los clientes dependa del nivel de educación.

6.4 CANTIDAD DE CLIENTES POR TIPO DE VIVIENDA

Se realizó un gráfico de barras con matplotlib que grafica la cantidad de clientes diferenciado por tipo de vivienda.

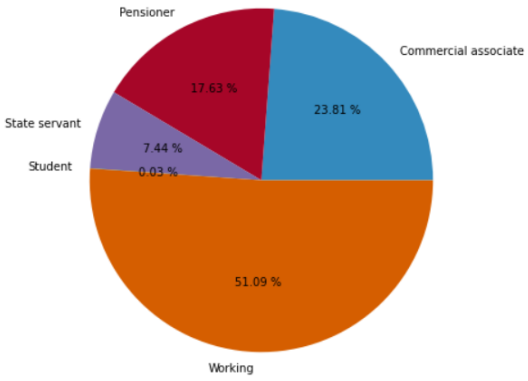


6.4.1 Conclusiones

- Se puede ver en la gráfica que la mayoría de los clientes viven en una casa o departamento.

6.5 CANTIDAD DE CLIENTES POR TIPO DE TRABAJO

Se realizó un gráfico de torta con matplotlib que grafica la cantidad de clientes diferenciado por tipo de ingreso.

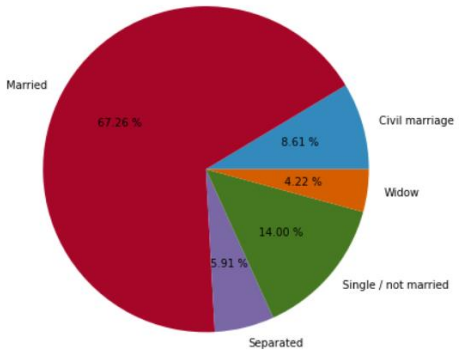


6.5.1 Conclusiones

- Más de la mitad de los clientes son empleados en relación de dependencia en el sector privado.
- El 24.30% son socios comerciales.
- Un porcentaje bajo de los clientes (0,03%) son estudiantes.

6.6 CANTIDAD DE CLIENTES POR TIPO DE ESTADO CIVIL

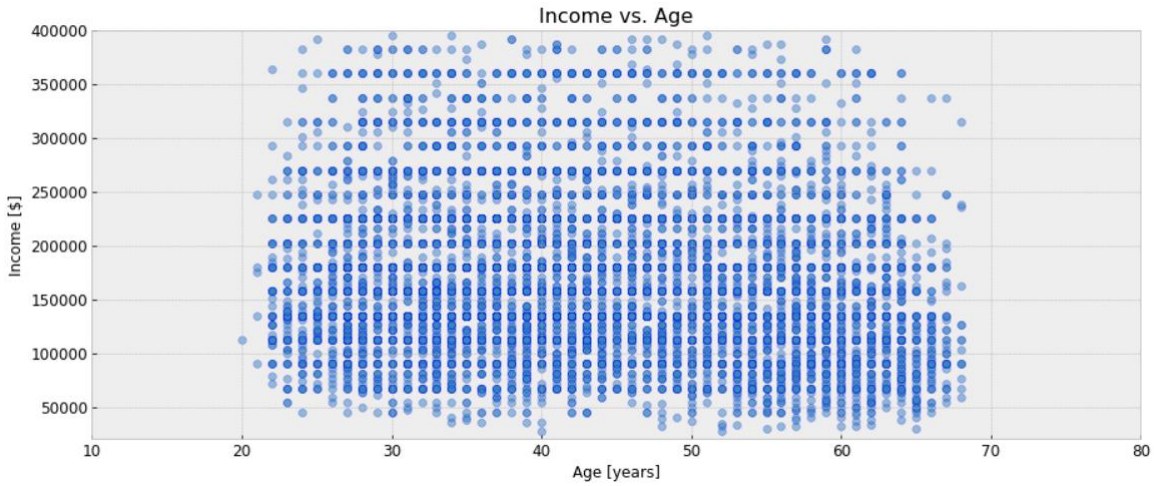
Se realizó un gráfico de torta con matplotlib que grafica la cantidad de clientes diferenciado por tipo de estado civil.



6.6.1 Conclusiones

- Se puede ver en la gráfica que la mayoría de los clientes (más del 68%) están casados.
- El 12.87% de los clientes se encuentra soltero.

6.7 INGRESOS DE LOS CLIENTES VS EDAD



6.7.1 Conclusiones

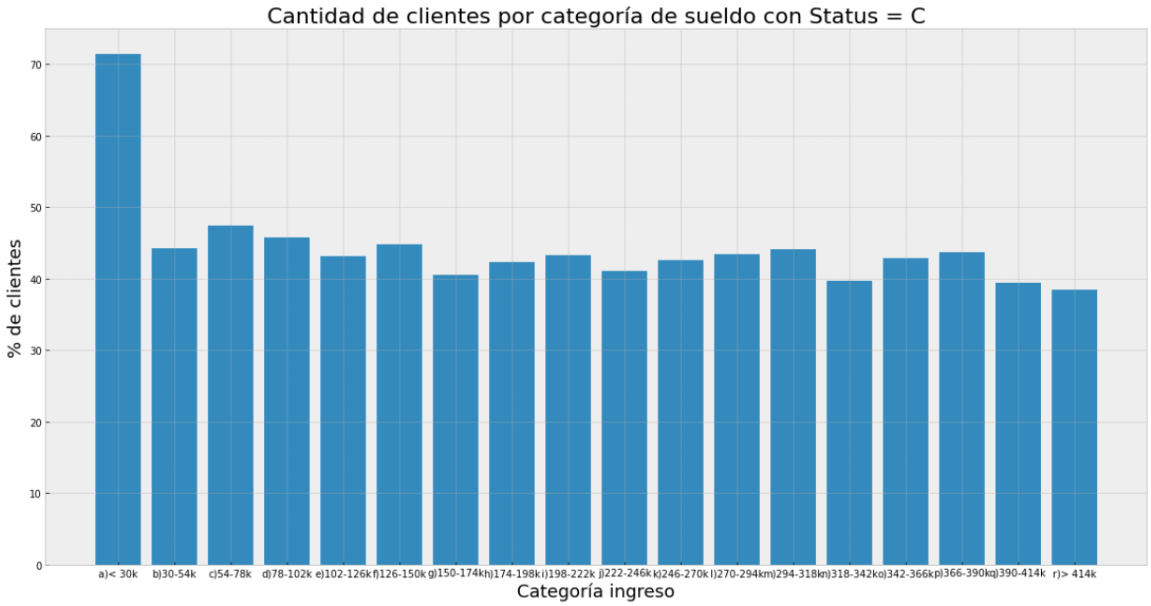
Como se observa en el gráfico no se pueden determinar subgrupos de clientes.

Por otro lado, se puede observar que la mayor parte de los clientes se agrupan entre ingresos de 50.000 usd y 200.000 usd y edades de 20 años y 65 años, ya que es la zona de mayor densidad en el gráfico.

6.8 INGRESOS DE LOS CLIENTES VS STATUS

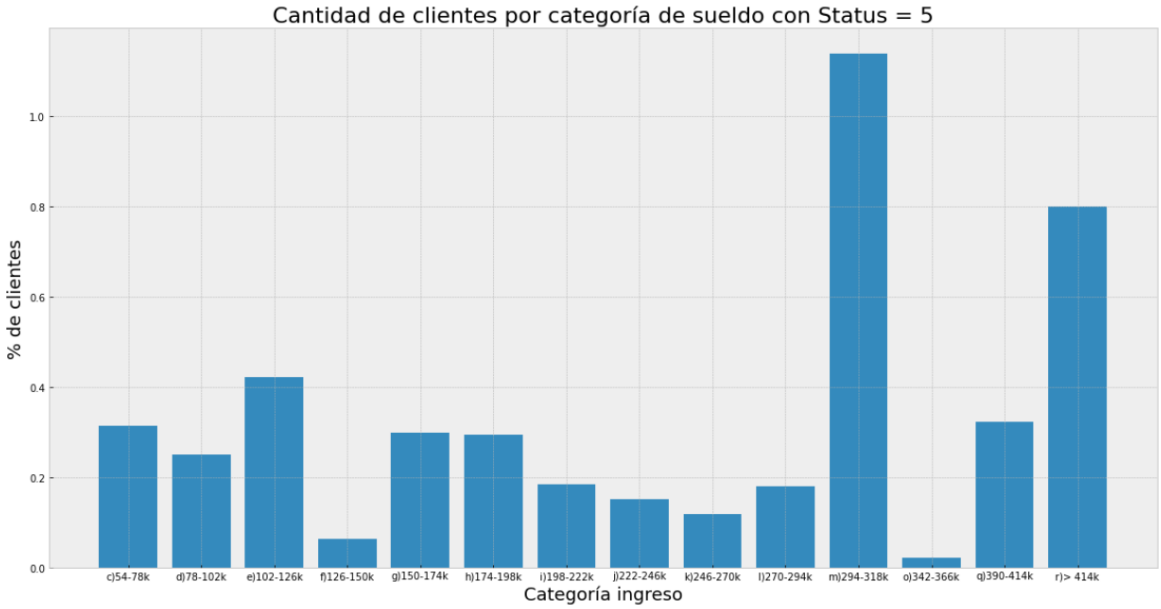
Para obtener los siguientes gráficos fue necesario previamente realizar el cálculo porcentual de clientes con los diferentes status, discriminado por rango de ingreso de los mismos.

6.8.1 Ingreso vs Status = C



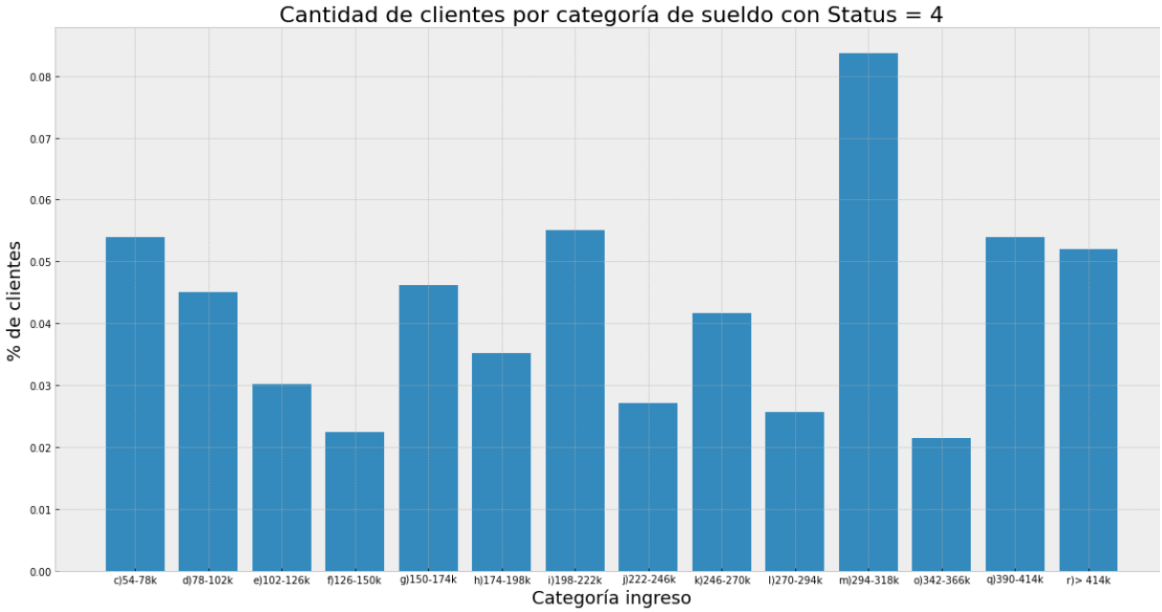
Conclusiones: los que ganan en el rango de menos de 30 k de ingreso tienen deudas canceladas en el mes y representan el 70% del total de clientes. También se puede concluir que de forma general independiente del rango de sueldo de los clientes más del 35% de los clientes tienen status= C.

6.8.2 Ingreso vs Status = 5



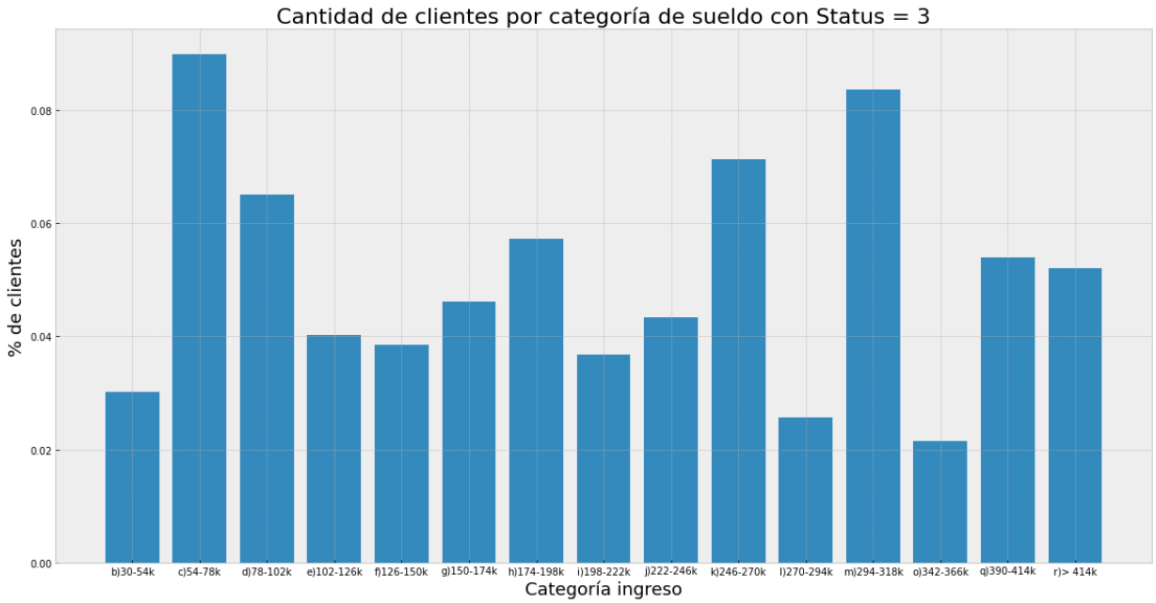
Conclusiones: los clientes con ingresos de 294 a 318 k son los que más tienen status = 5 (deudas incobrables o canceladas luego de 150 días) y superan el 1% de los clientes en ese rango de ingreso.

6.8.3 Ingreso vs Status = 4



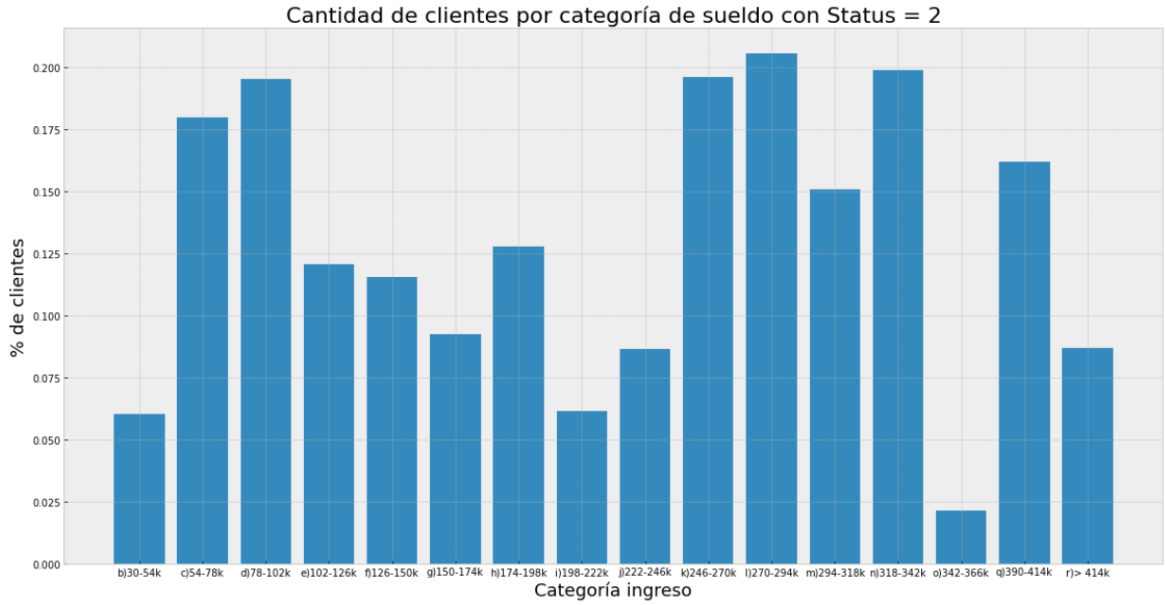
Conclusiones: los clientes con ingresos de 294-318 k son los que más tienen status = 4, y no superan el 1% de los clientes en ese rango de ingreso.

6.8.4 Ingreso vs Status = 3



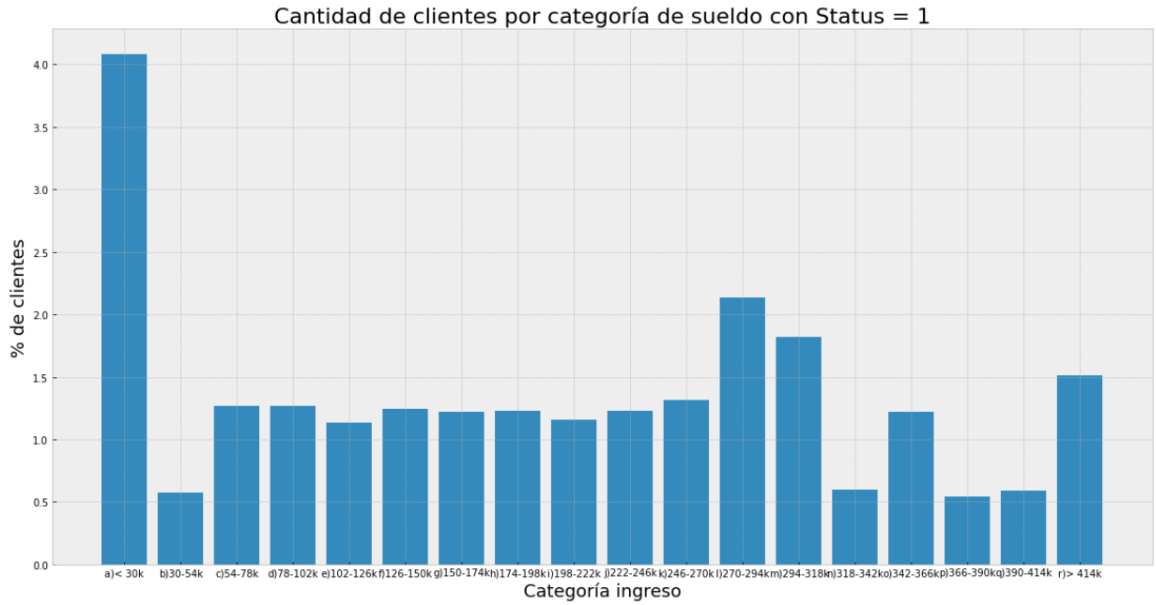
Conclusiones: los clientes con ingresos de 54 a 78 k y 294 a 318 k son los que más tienen deudas de 90 a119 días de atraso, y no superan el 1% de los clientes en ese rango de ingreso.

6.8.5 Ingreso vs Status = 2



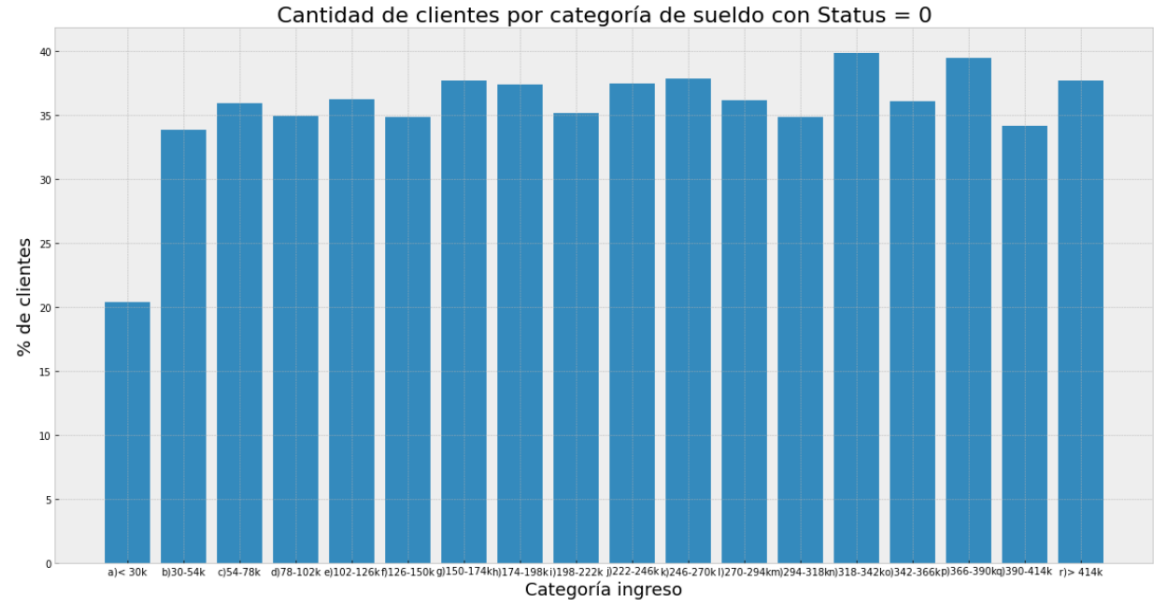
Conclusiones: los clientes que tienen status = 2 (60-89 días de atraso), son una muestra muy pequeña, que no superan el 0.2%.

6.8.6 Ingreso vs Status = 1



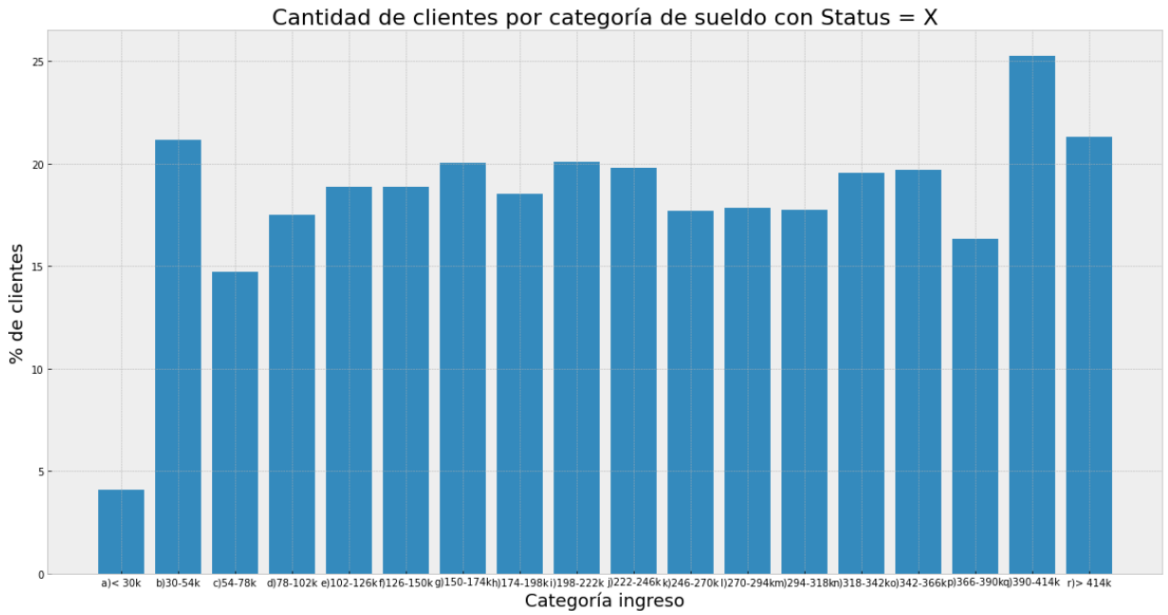
Conclusiones: los clientes que con ingresos de menos de 30 k son los que en porcentaje tienen status= 1 (30-59 días de atraso), y son el 4% de los clientes en ese rango de ingresos.

6.8.7 Ingreso vs Status = 0



Conclusiones: el 20% de los clientes con ingresos de menos de 30 k tienen status = 0 (1-29 días de atraso), entre los ingresos de 30 k a más de 414 k superan siempre el 30% de los clientes con este status.

6.8.8 Ingreso vs Status = X



Conclusiones: los clientes que porcentualmente tienen status=X (sin prestamos en el mes) son los que tienen ingresos entre 390 a 414 k, y representan el 25% de la totalidad de los clientes en ese rango de ingresos.

7 PREPARAR DATOS PARA EL MODELO

Mediante la implementación de modelos, se busca identificar los clientes de la base de datos que deben ser aceptados o rechazados por el banco, para recibir un préstamo. Se considerará como clientes aceptados y rechazos a los que cumplan con el siguiente status:

- Cliente aceptado: cuando su status sea C o 0.
- Cliente no aceptado: cuando su status sea 1, 2, 3, 4 o 5.

Para obtener el dataframe con el cual entrenar el modelo y luego testearlo se realizaron los siguientes pasos:

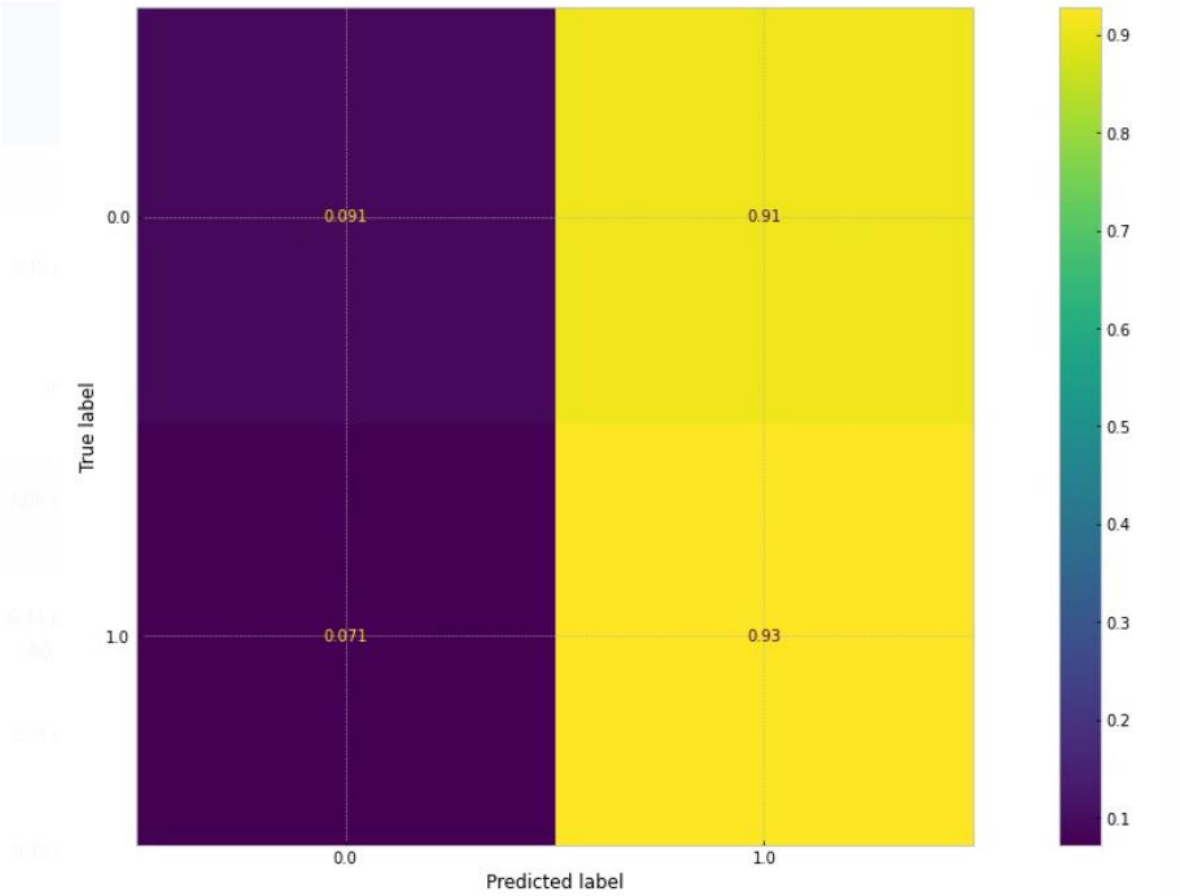
1) Eliminar los clientes cuyo status sea X, ya que estos son los clientes que nunca obtuvieron un préstamo y por lo tanto no se tiene información crediticia.
2) Identificar los clientes aceptados con 1 y los no aceptados con 0.
3) Obtener un listado con los clientes únicos.
4) Identificar la distribución de las variables categóricas.
5) Realizar One Hot Encoding para las variables categóricas no ordinales.
6) Realizar un Ordinal Encoding a las variables categóricas ordinales.
7) Eliminar las columnas que no aportan valor al dataframe:
    - Variables categóricas, ya que luego de realizar el One Hot Encoding, estas son identificadas por las nuevas variables arrojadas por este método.
    - Se eliminan las siguientes columnas, ya que no son útiles para el modelo: 'id', 'amt_income_total', 'days_birth', 'data_day_x', 'birthday', 'months_balance', 'status', 'months_balance_date', 'edad_dias', 'amt_income_total_outliers','cnt_children_outliers', 'cnt_fam_members_outliers'.
8) Realizar la Normalización de las variables del modelo, para que las mismas estén entre 0 y 1.
9) Establecer las variables independientes y la variable objetivo del modelo.
10) Verificar el balanceo de la variable objetivo, en donde se observa que la cantidad de clientes aceptados es de un 82,7% y los clientes no aceptados son un 17,3%. Esto indica que la variable objetivo está desbalanceada.
11) Realizar el análisis de balanceo de la variable objetivo del dataframe test, para verificar que es similar a la del dataframe de entrenamiento.

8 SELECCIÓN DEL MODELO

A continuación, se entrenan los siguientes modelos para predecir la variable objetivo.

8.1 KNN – VECINOS CERCANOS

Porcentaje de aciertos sobre el set de evaluación de vecinos cercanos (accuracy): 0.784.

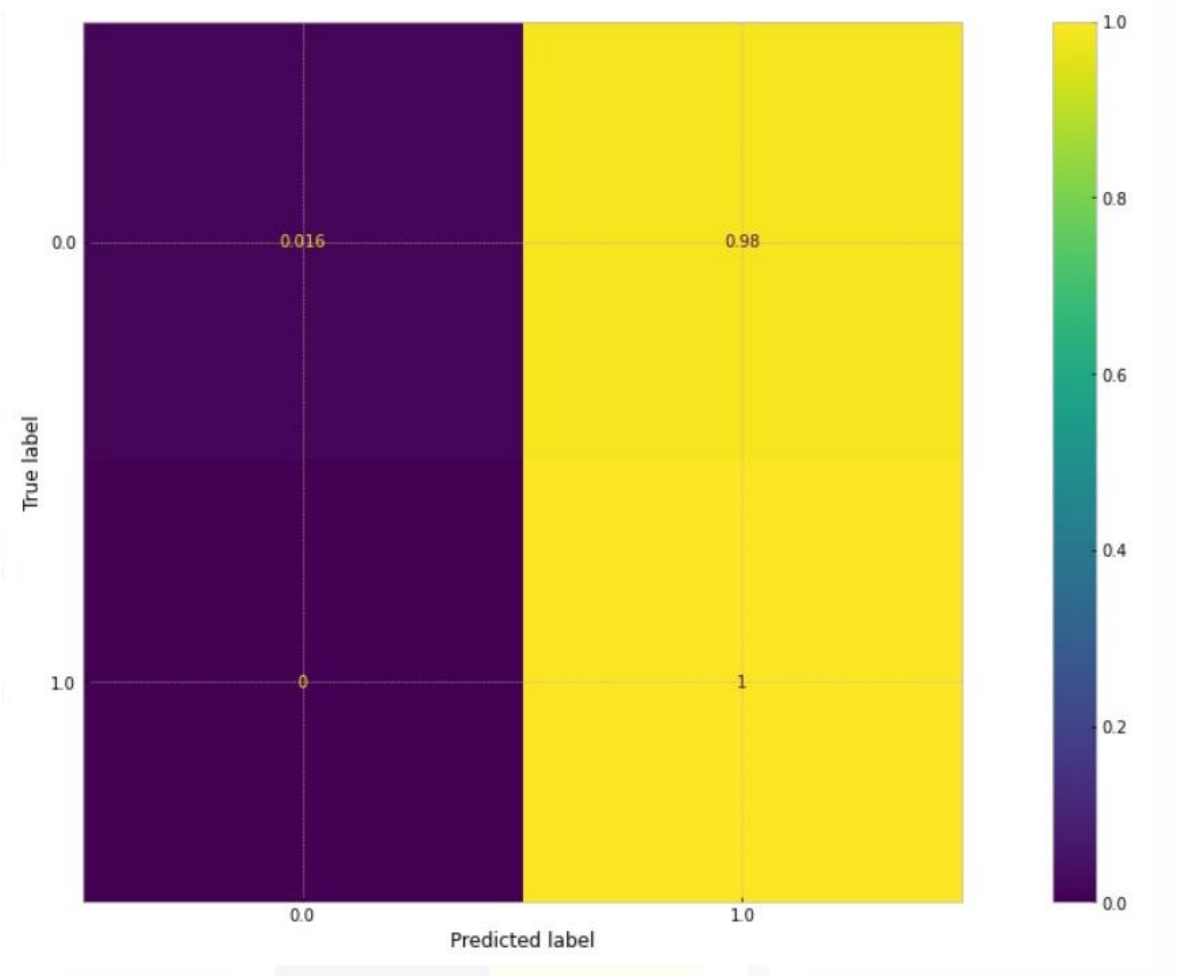



```
precision 0.83
recall 0.93
f1_score 0.88
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados. Esto quiere decir, que el modelo comete errores de tipo 2.

8.2 REGRESIÓN LOGÍSTICA

Porcentaje de aciertos sobre el set de evaluación de regresión logística (accuracy): 0.830.

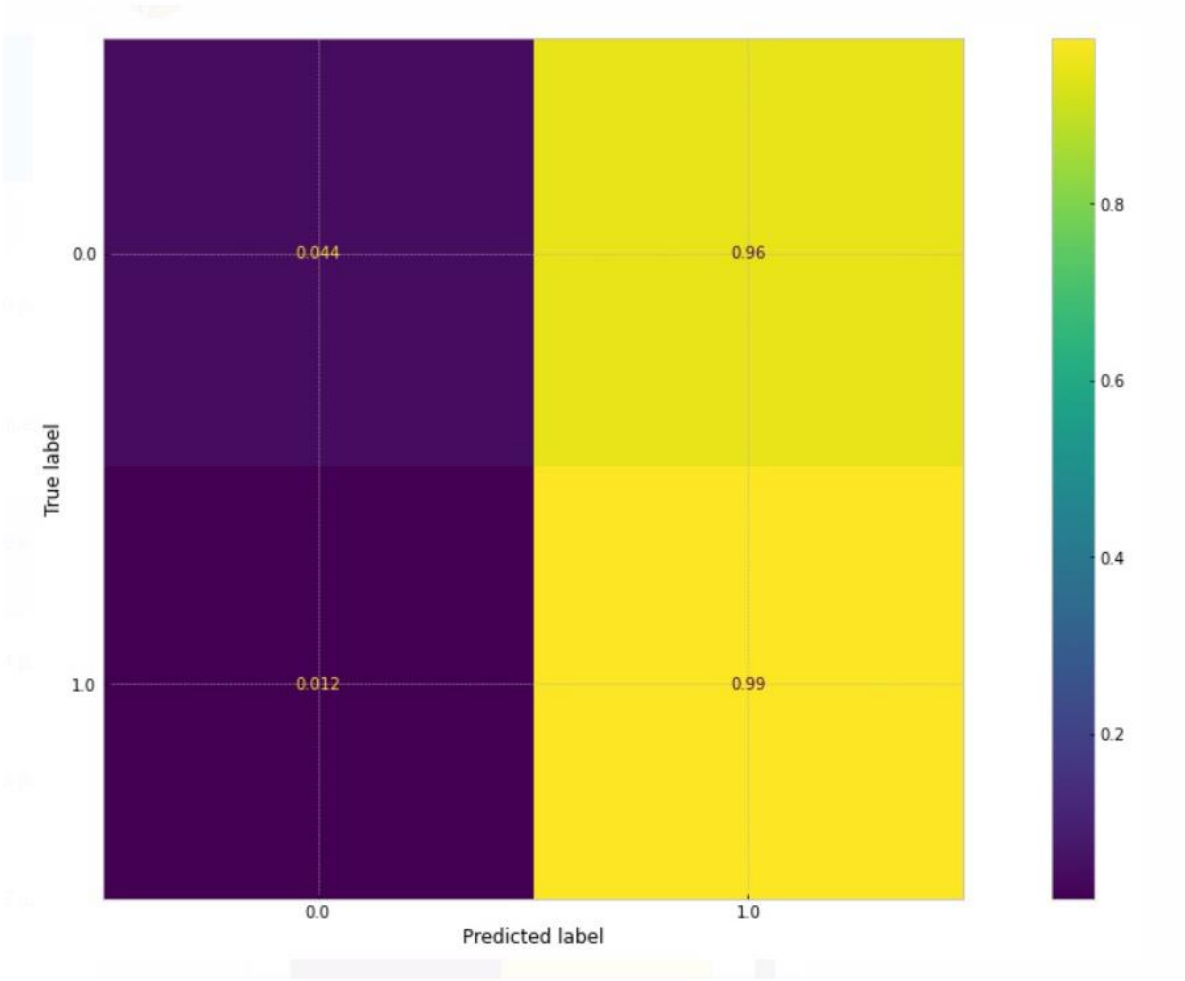



```
precision 0.8297394429469901
recall 1.0
f1_score 0.9069481954333415
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados. Esto quiere decir, que el modelo comete errores de tipo 2.

8.3 RANDOM FOREST

Porcentaje de aciertos sobre el set de evaluación de random forest (accuracy): 0.824.




```
precision 0.8297394429469901
recall 1.0
f1_score 0.9069481954333415
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados. Esto quiere decir, que el modelo comete errores de tipo 2.

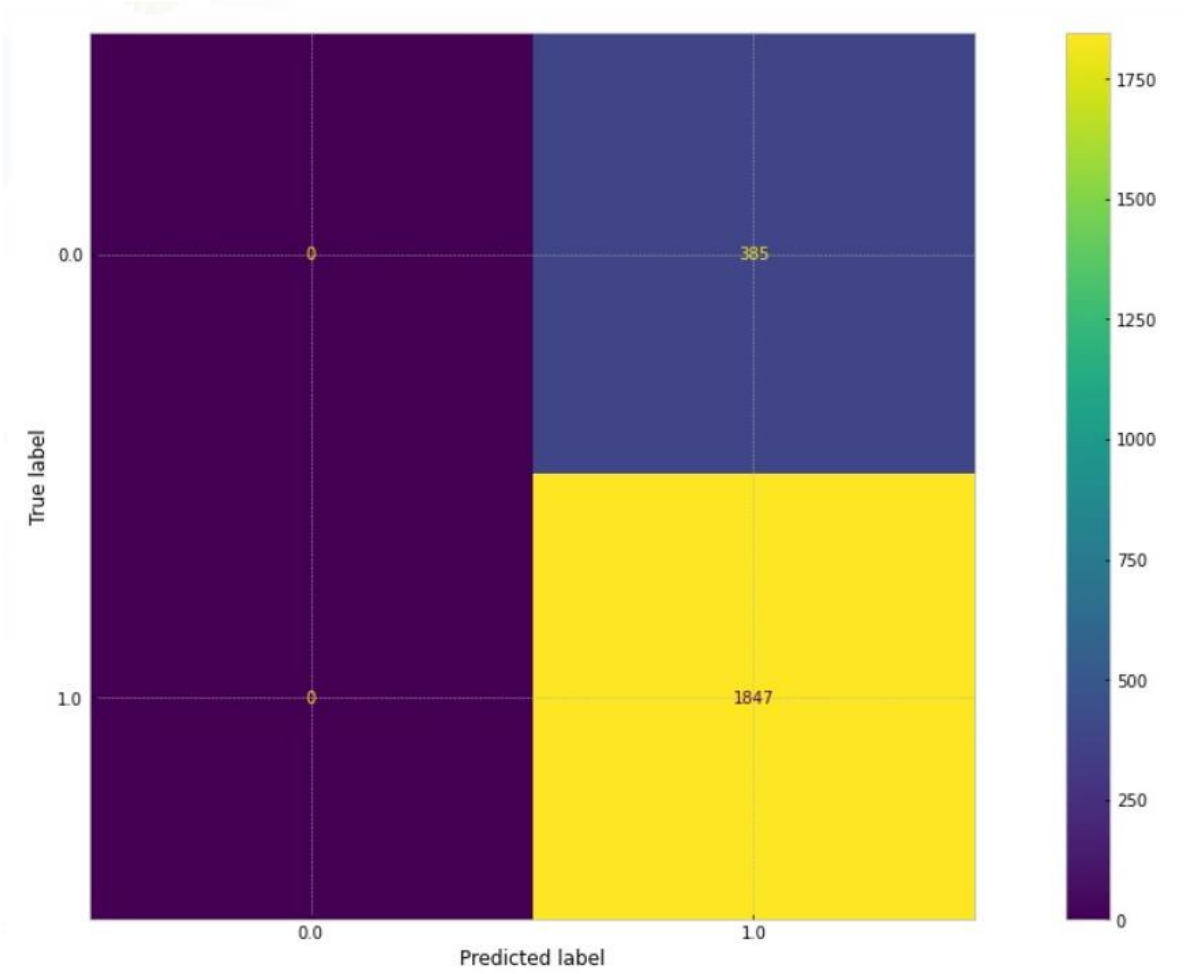
8.4 REDUCCIÓN DE VARIABLES PARA APLICAR NUEVOS MODELOS

Luego de aplicar los algoritmos, se evalúa como influyen las variables en el modelo y se crea un nuevo dataframe en donde solo se tienen en cuenta aquellas que mayor influencia tienen.

```
Importancia de características:
Característica edad (0.197710)
Característica days_employed (0.182147)
Característica categoria_sueldo (0.145538)
Característica cnt_fam_members (0.052598)
Característica name_education_type (0.040737)
Característica flag_own_car_Y (0.031451)
Característica cnt_children (0.030838)
Característica flag_phone_True (0.028899)
Característica name_income_type_Working (0.027161)
Característica flag_own_realty_Y (0.026985)
Característica code_gender_M (0.025996)
Característica flag_work_phone_True (0.024282)
Característica occupation_type_Not informed (0.015655)
Característica occupation_type_Laborers (0.015564)
Característica flag_email_True (0.015242)
Característica occupation_type_Core staff (0.012257)
Característica name_housing_type_House / apartment (0.011527)
Característica name_income_type_State servant (0.011442)
Característica occupation_type_Managers (0.011303)
Característica occupation_type_Sales staff (0.010724)
Característica occupation_type_Drivers (0.009711)
Característica occupation_type_High skill tech staff (0.007819)
Característica name_housing_type_With parents (0.007172)
Característica name_income_type_Pensioner (0.006990)
Característica occupation_type_Medicine staff (0.006863)
Característica name_housing_type_Municipal apartment (0.006123)
Característica occupation_type_Cooking staff (0.005515)
Característica occupation_type_Security staff (0.005330)
Característica name_housing_type_Rented apartment (0.004604)
Característica occupation_type_Cleaning staff (0.004078)
Característica occupation_type_Low-skill Laborers (0.003733)
Característica occupation_type_Not working (0.002957)
Característica occupation_type_Private service staff (0.002909)
Característica name_housing_type_Office apartment (0.002656)
Característica occupation_type_Waiters/barmen staff (0.001413)
Característica occupation_type_Secretaries (0.001151)
Característica occupation_type_HR staff (0.001128)
Característica occupation_type_Realty agents (0.000689)
Característica name_income_type_Student (0.000550)
Característica occupation_type_IT staff (0.000549)
```

8.5 REGRESIÓN LOGÍSTICA (CON REDUCCIÓN DE VARIABLES):

Porcentaje de aciertos sobre el set de evaluación de regresión logística (accuracy): 0.827.



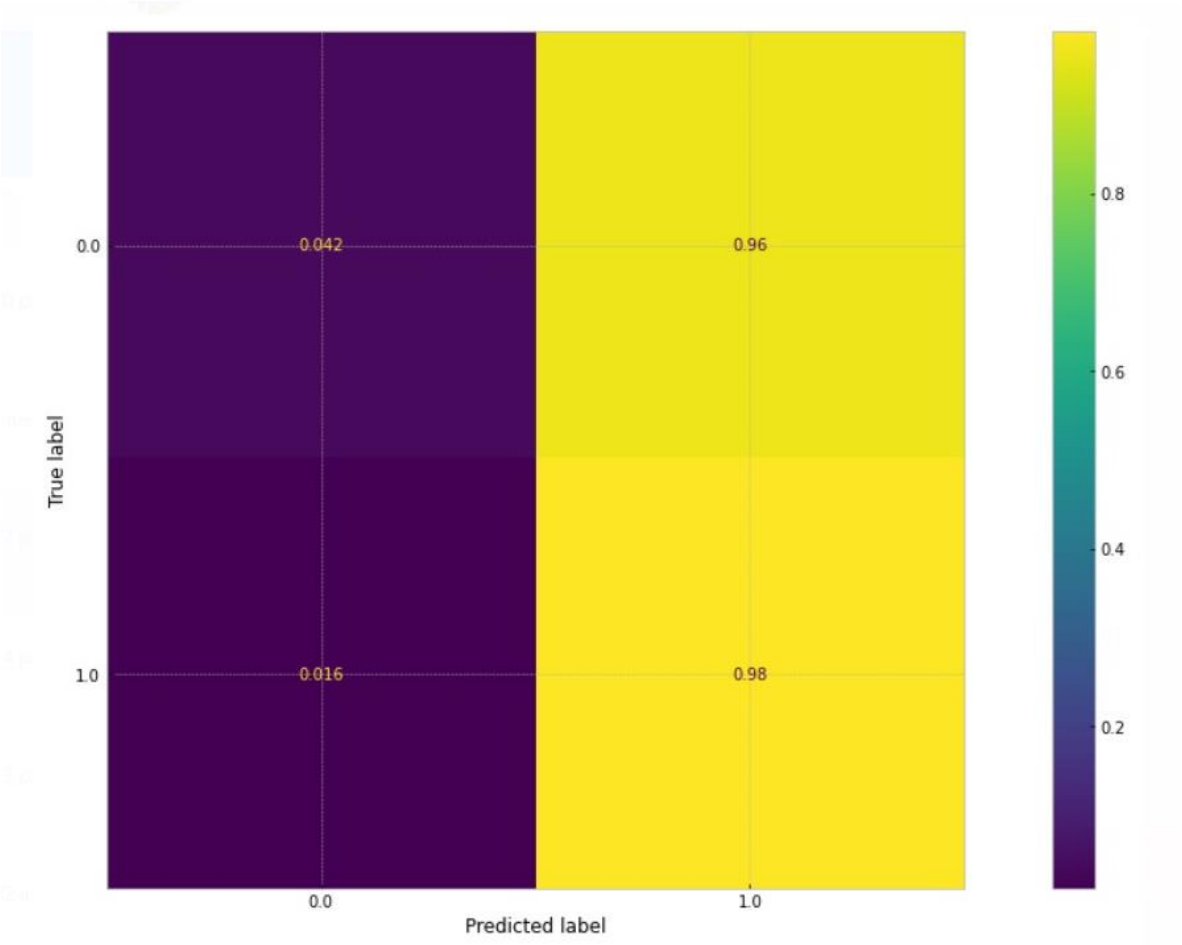

```
precision 0.83
recall 1.0
f1_score 0.91
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados. Esto quiere decir, que el modelo comete errores de tipo 2.

Por otro lado, se puede observar que el modelo no obtuvo mejoras al reducir la cantidad de variables, sino que el accuracy empeoró.

8.6 RANDOM FOREST (CON REDUCCIÓN DE VARIABLES)

Porcentaje de aciertos sobre el set de evaluación de random forest (accuracy): 0.821.




```
precision 0.83
recall 0.98
f1_score 0.9
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados. Esto quiere decir, que el modelo comete errores de tipo 2.

Por otro lado, se puede observar que el modelo no obtuvo mejoras al reducir la cantidad de variables, sino que el accuracy empeoró

8.7 CONCLUSIÓN:

Como se puede observar los modelos utilizados predicen bien cuando un cliente debe ser aceptado, pero cometen muchos errores de tipo 2, es decir arrojan muchos falsos positivos.

Al disminuir la cantidad de variables que ingresan a los modelos mediante no se observa ninguna mejora en el desempeño.

Finalmente, luego de analizar la estructura de los datos se concluye que los modelos no están prediciendo bien la variable objetivo ya que esta se encuentra desbalanceada y se deberían implementar métodos que permitan trabajar con este tipo de dataframe.

9 DESCARGA DE DATOS DESDE APIS PÚBLICAS

En el siguiente apartado, se obtienen datos sobre la economía de Estados Unidos desde la siguiente APIs https://www.econdb.com/api/series/.

Esta API está compuesta por más de 6000 variables, de las cuales se seleccionan las siguientes:

- Unemployment.
- Consumer price index (ipc).
- Utilization rate
- Commercial balance (goods + services).
- Retail trade
- Industrial production
- Consumer confidence index
- Producer price index
- Government debt
- House Price
- Sentiment index
- Population
- Money supply
- Policy rate - short term

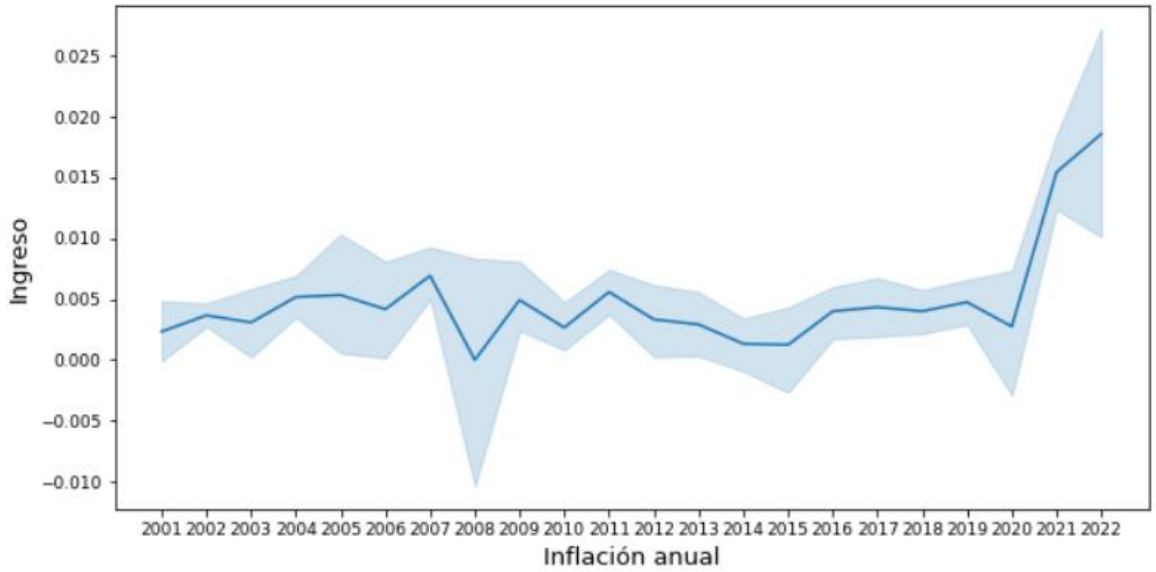
10 DATAWRANLING DE APIS

En este este apartado se realiza la limpieza de datos del dataframe economy (obtenido a partir de APIs) y se llevan todas las variables a las unidades correctas. Luego de esto, se normalizan las variables del dataset.

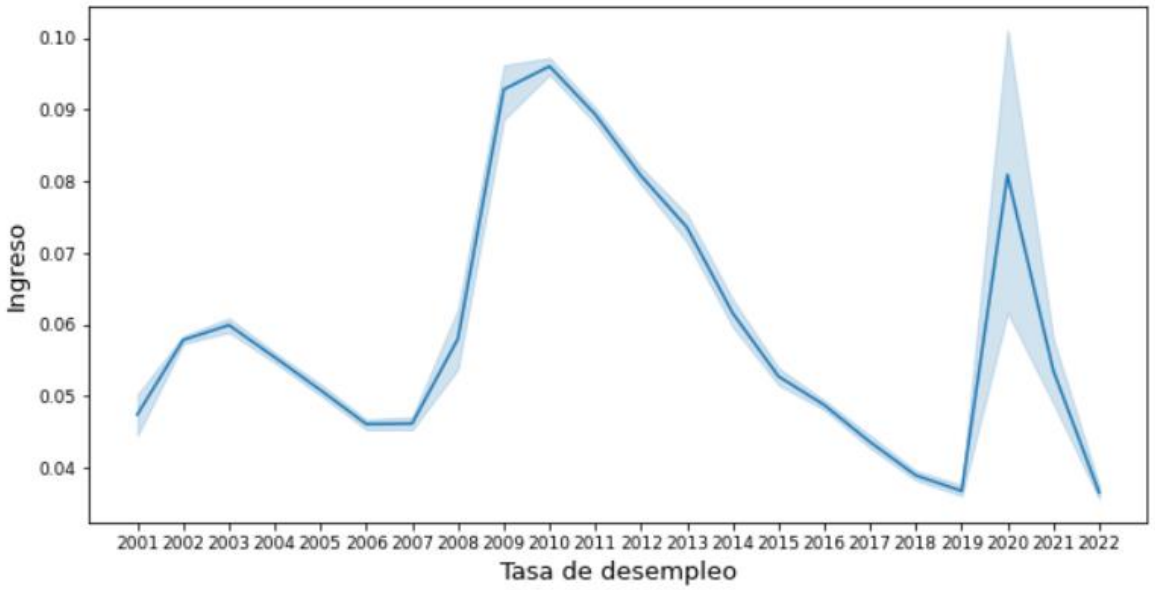
Finalmente, se realiza una unión entre el dataframe original del proyecto y el dataset economy mediante la columna anio_mes, obteniedose df_economy.

10.1 EJEMPLOS GRÁFICOS API

10.1.1 Inflación anual a lo largo del tiempo



10.1.2 Tasa de desempleo a lo largo del tiempo



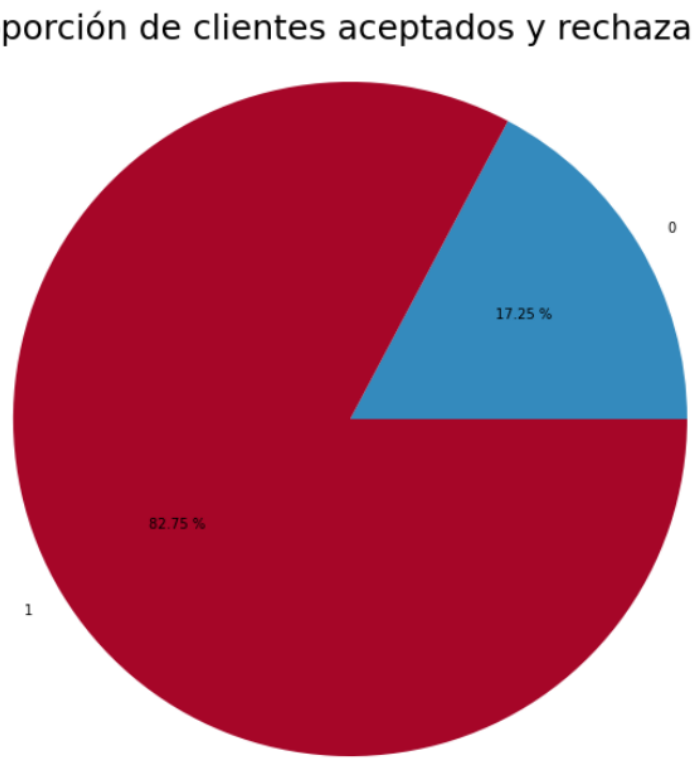
11 STORITELLYNG

Para entregar una tarjeta de crédito o un préstamo los bancos deben evaluar la situación crediticia de los clientes, para lo cual solicitan el servicio de una consultora encargada de evaluar el riesgo crediticio.

Las evaluaciones de riesgos crediticios son un método común de control de riesgos en la industria financiera. La información personal y los datos presentados por los solicitantes de tarjetas de crédito es utilizada para predecir la probabilidad de futuros incumplimientos y préstamos de tarjetas de crédito. A partir de esta información, el banco puede decidir si emite una tarjeta de crédito al solicitante, ya que los puntajes de crédito pueden cuantificar objetivamente la magnitud del riesgo.

La pregunta que hay que realizarse es ¿cuáles son las variables que mayormente explican a mis clientes rechazados?, para esto, se analizaron los siguientes atributos:

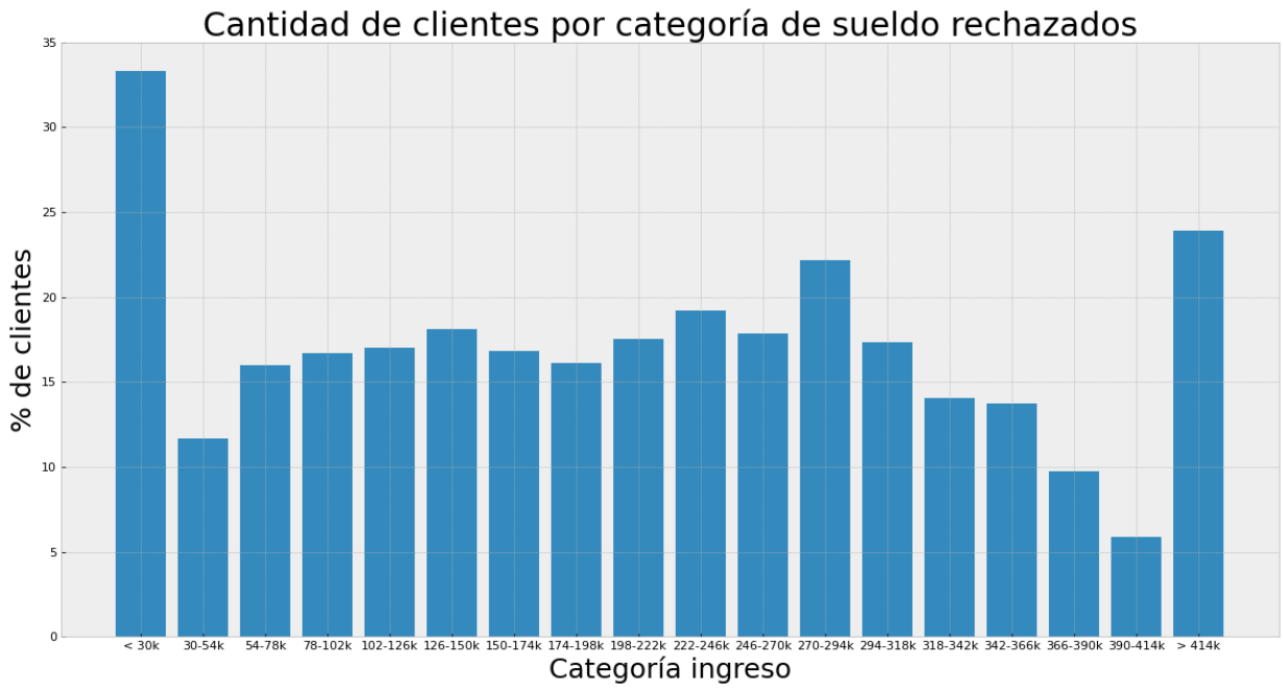
11.1 PROPORCIÓN DE CLIENTES ACEPTADOS Y RECHAZADOS



Referencia: 0, cliente rechazado. 1, cliente aceptado.

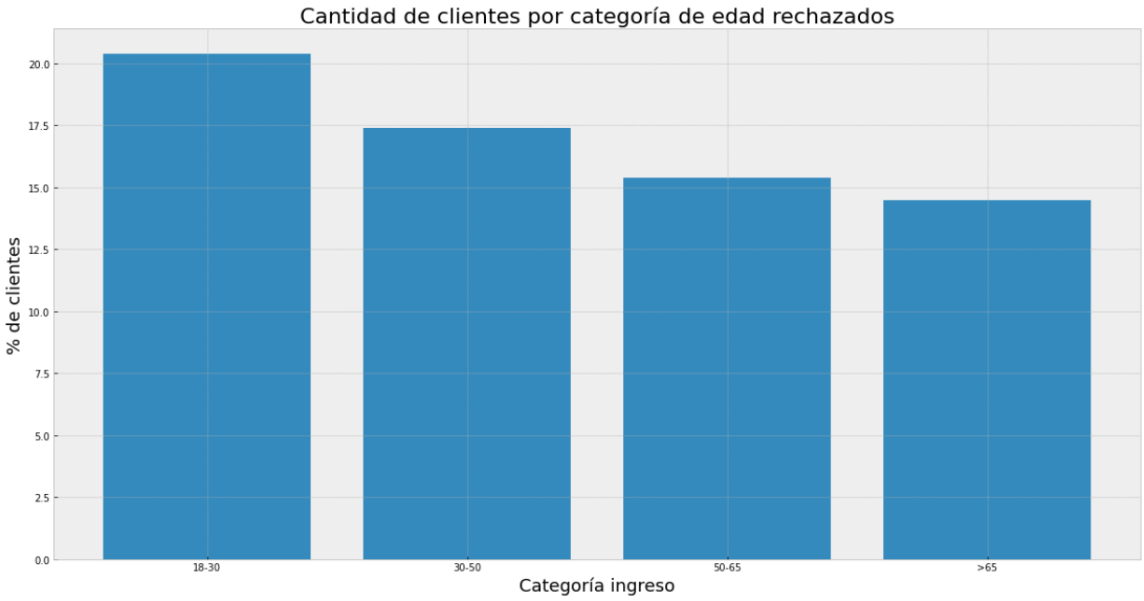
Como se puede obtener de la gráfica, el dataset que contiene la información de los clientes se encuentra desbalanceado, ya que la proporción de clientes aceptados es mucho mayor que la de rechazados.

11.2 INGRESO DE LOS CLIENTES RECHAZADOS



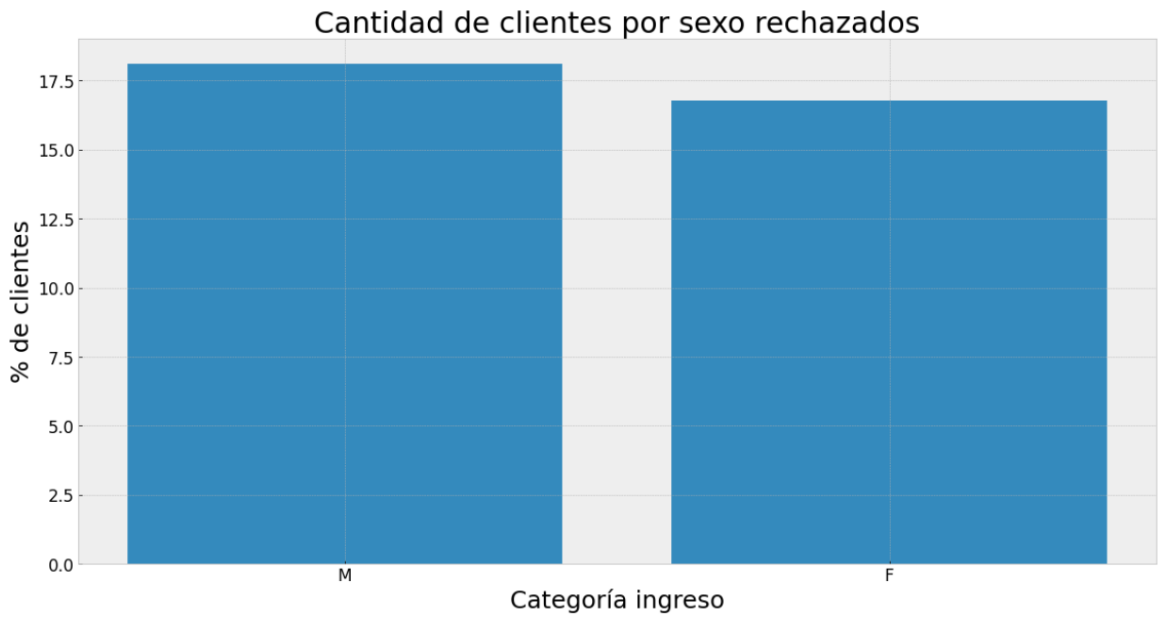
De esta gráfica se puede observar que los clientes que más rechazos obtienen son los de los extremos, es decir, los que más y menos ingresos anuales tienen.

11.3 RANGO DE EDAD DE LOS CLIENTES RECHAZADOS



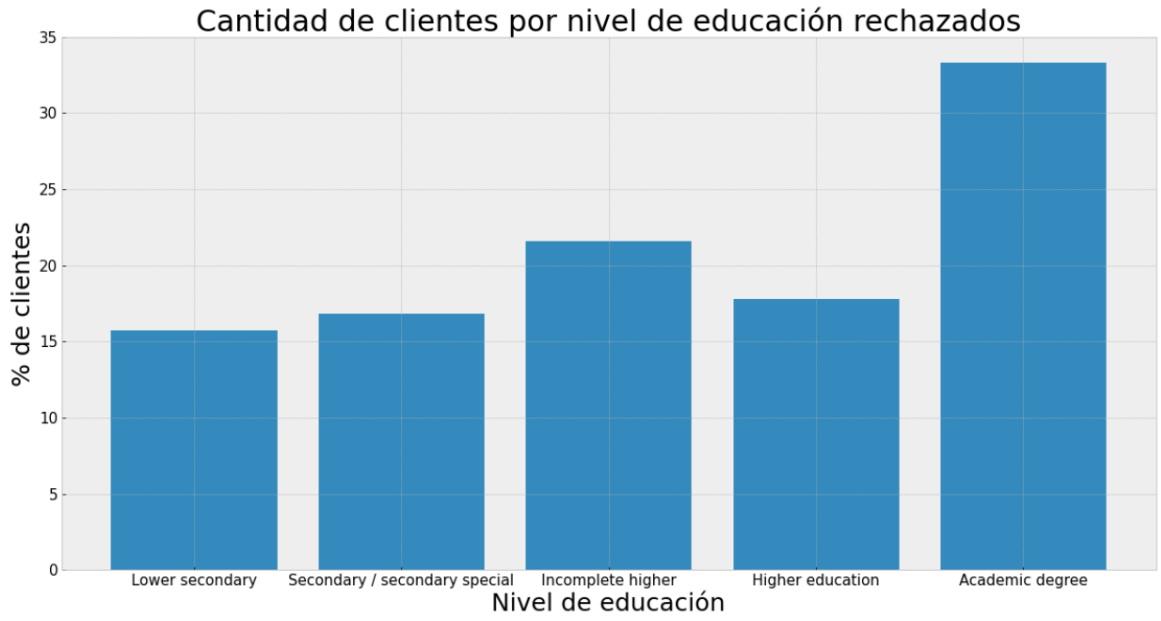
De esta gráfica se puede observar que a medida que aumenta la edad de los clientes disminuyen los rechazos.

11.4 GENERO DE LOS CLIENTES RECHAZADOS



Del gráfico se puede observar que los clientes del género masculino son los que mayores rechazos reciben.

11.5 CLIENTES RECHAZADOS POR NIVEL DE EDUCACIÓN



Del gráfico se observa que a mayor nivel de educación aumentan los clientes rechazados. Por otro lado, se debe destacar que en la categoría “Academic degree” se observa un alto nivel de rechazados porque el tamaño de la muestra es pequeño y al no ser representativo, no se debería obtener conclusiones de esta categoría.

Finalmente, debido a que la variable objetivo es una variable categórica, luego de este análisis exploratorio, se recomienda implementar el uso de modelos del tipo categórico. Por otro lado, se tiene un dataset desbalanceados por lo que se deberán aplicar técnicas de balanceo.

12 ENTRENANDO UN ALGORITMO DE MACHINE LEARNING

Luego de haber adquirido nuevas herramientas para optimizar modelos, se procede a reentrenar los mismos, con el objetivo de obtener mejores métricas resultantes.

12.1 INGENIERÍA DE VARIABLES

A continuación, se crearán las siguientes variables que pueden resultar útiles para entrenar el modelo:

- categoria_sueldo: existe una gran variabilidad entre los ingresos de los clientes del dataset y esta no corresponde a datos mal cargados. Para evitar que estos generen problemas a la hora del entrenamiento del modelo se generaron categorías de sueldos.
- categoria_edad: esta variable se creo con el objetivo de observar si existen grupos etarios (jóvenes, adultos, ancianos) que posean mayor cantidad de rechazos.
- tiene_hijos: esta variable se creó ya que en el dataframe original no existía ninguna variable que permitiese identificar de manera directa si un cliente tiene o no tiene hijos.
- vive_solo: esta variable se creó ya que en el dataframe original no existía ninguna variable que permitiese identificar de manera directa si un cliente vive solo o en familia.

- alquila: esta variable se creó ya que en el dataframe original no existía ninguna variable que permitiese identificar de manera directa si un cliente es dueño de una vivienda o alquila.
- trabajando: esta variable se creó ya que en el dataframe original no existía ninguna variable que permitiese identificar de manera directa si un cliente se encuentra trabajando o está desempleado.

12.2 PROCESOS DE ENCODING

Para poder entrenar algunos modelos es necesario transformar las variables categóricas en variables numéricas. Para esto se aplican las siguientes técnicas:

- One Hot Encoding: se utiliza para variables no ordinales con pocas categorías. Esto se realizó en las siguientes columnas del dataset: 'code_gender', 'flag_own_car', 'flag_own_realty', 'flag_mobil', 'flag_work_phone', 'flag_phone', 'flag_email'.
- Label Encoding: se utiliza para variables no ordinales con muchas categorías. Esto se realizó en las siguientes columnas del dataset: 'name_income_type','name_housing_type','occupation_type', 'categoria_edad', 'name_family_status'.
- Ordinal Encoding: se utiliza para variables ordinales. Esto se realizó en las siguientes columnas: 'name_education_type', 'categoria_sueldo'.

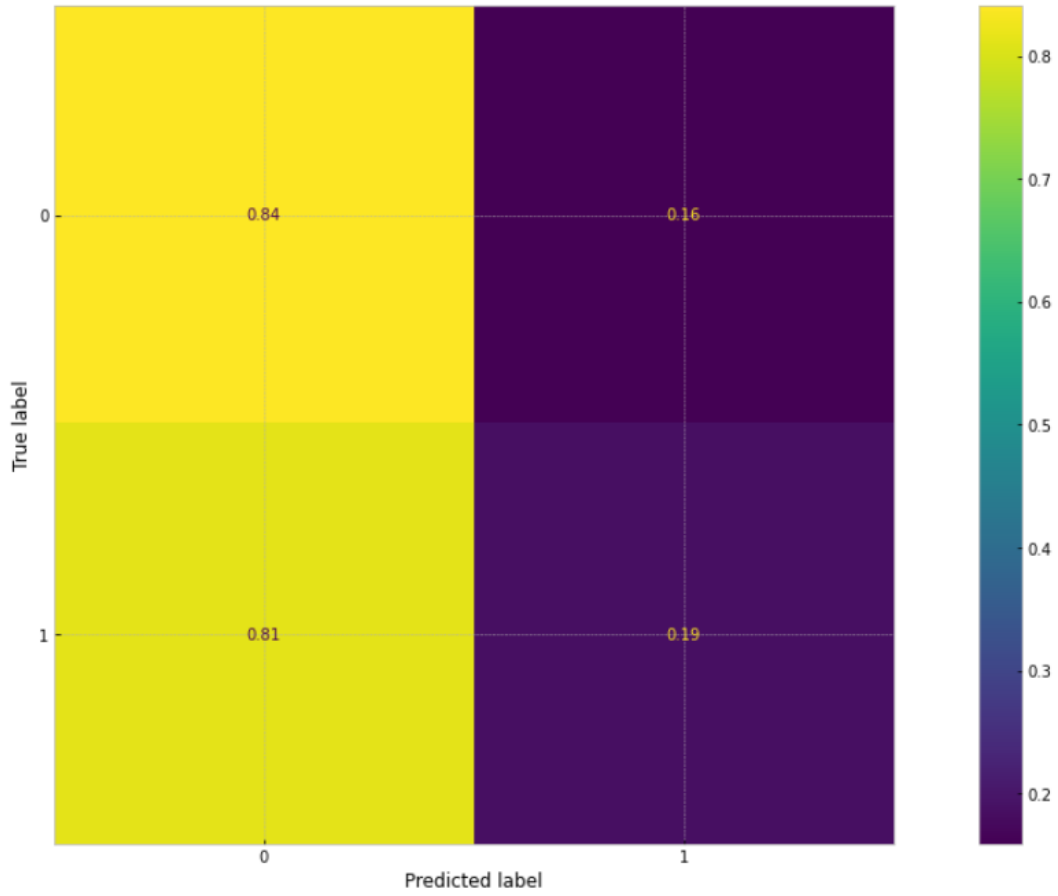
Finalmente, para que todas las variables numéricas sean comparables entre sí, se deben normalizar con MinMax Scaler las siguientes columnas del dataset:'cnt_children','name_education_type','cnt_fam_members','name_income_type_label','name_housing_type_label','occupation_type_label','categoria_edad_label','name_family_status_label','categoria_sueldo'.

12.3 ENTRENAMIENTO DE MODELOS

A continuación, se entrenan los siguientes modelos para predecir la variable objetivo.

12.3.1 Regresión logística

Porcentaje de aciertos sobre el set de evaluación de regresión logística (accuracy): 0.85.

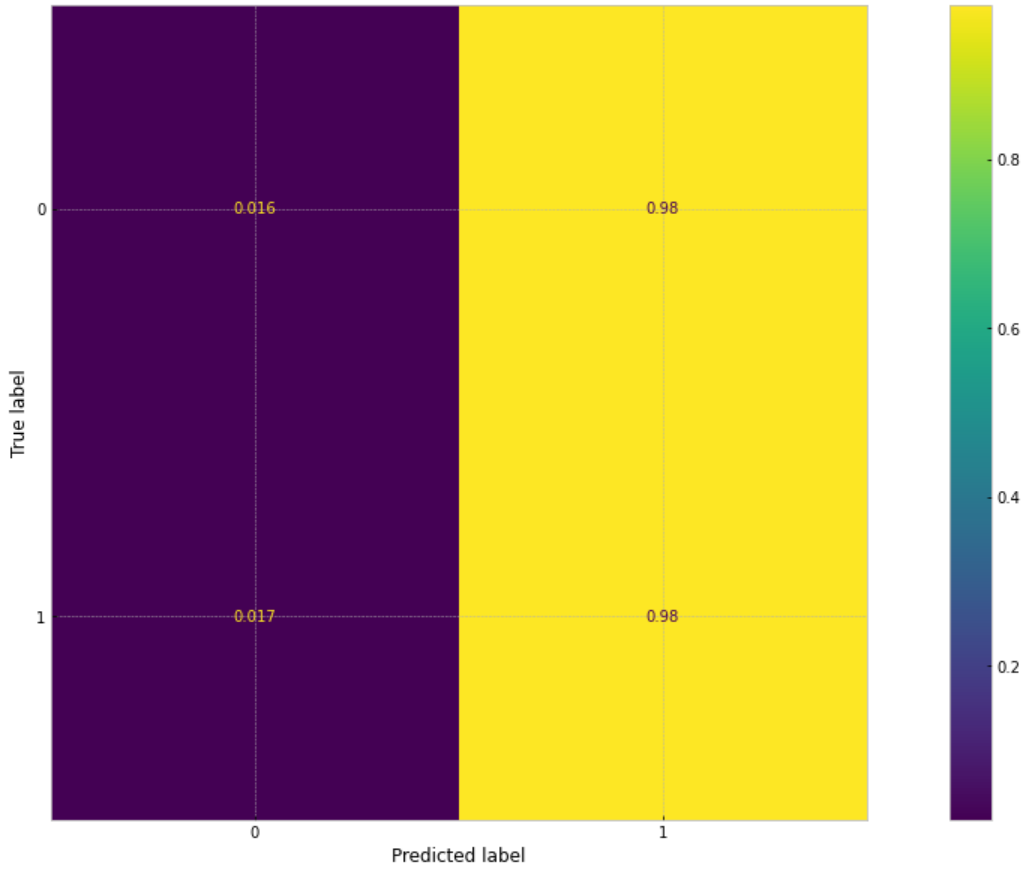



```
precision 0.85
recall 0.19
f1_score 0.31
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes rechazados, pero no es bueno para predecir los clientes que deben ser aceptados.

12.4 RANDOM FOREST

Porcentaje de aciertos sobre el set de evaluación de random forest (accuracy): 0.83.

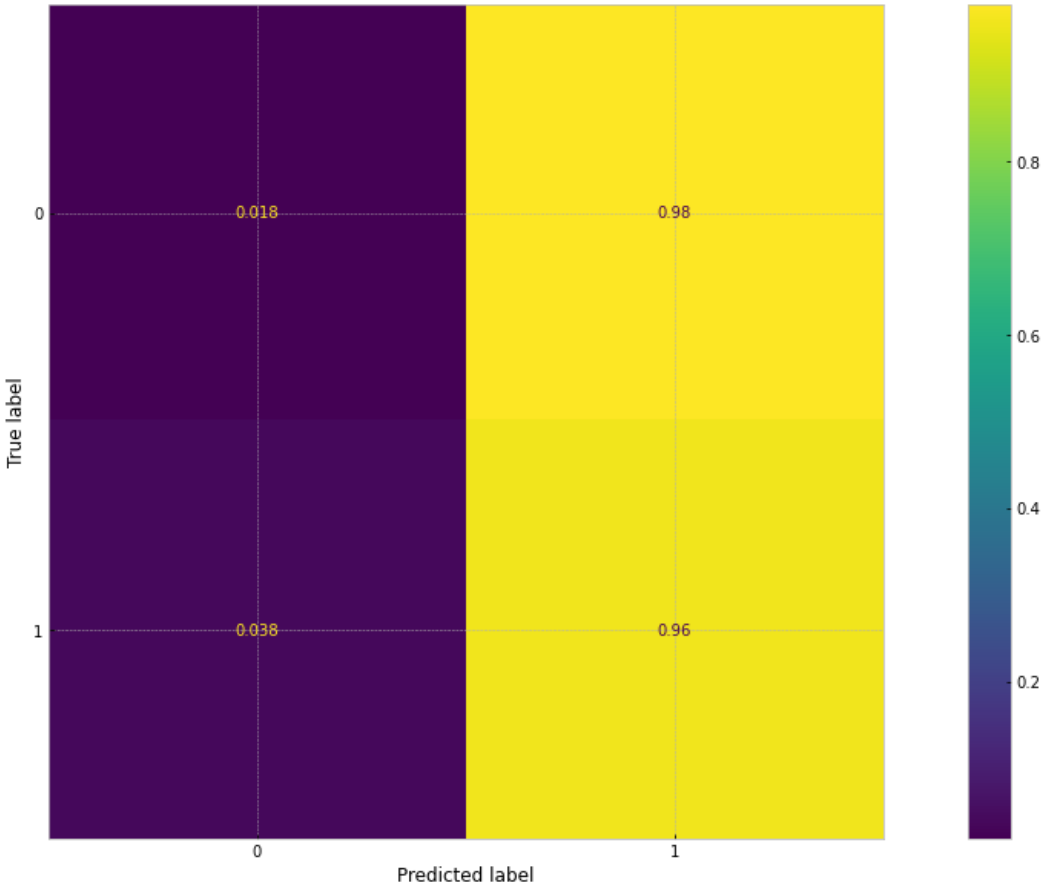



```
precision 0.83
recall 0.98
f1_score 0.9
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados.

12.5 SVM

Porcentaje de aciertos sobre el set de evaluación de SVM (accuracy): 0.79.




```
precision 0.82
recall 0.96
f1_score 0.89
```

Como se puede observar en la matriz de confusión y métricas calculadas, el modelo es bueno prediciendo los clientes aceptados, pero no es bueno para predecir los clientes que deben ser rechazados.

12.6 CONCLUSIÓN:

El modelo de Regresión Logística es bueno prediciendo los clientes que deberían ser rechazados, pero no es efectivo prediciendo a los clientes que deberían ser aceptados.

Los modelos de Random Forest y SVM predicen bien cuando un cliente debería ser aceptado, pero cometen muchos errores de tipo 1, es decir arroja muchos falsos positivos.

Por otro lado, se entrenaron todos los modelos con un dataset sin outilers y los resultados obtenidos fueron similares a los de los modelos que se entrenaron con el dataframe original. Luego de analizar los datos outliers, se verificó que estos son datos reales (es decir, no fueron introducidos por error de tipeos). Por este motivo, los nuevos modelos y la optimización de estos se harán con el dataset original.

Luego de analizar la estructura de los datos se concluye que los modelos no están prediciendo bien la variable objetivo ya que esta se encuentra desbalanceada y se deberían implementar métodos que permitan trabajar con este tipo de dataframe.

Finalmente, también se recomienda optimizar el uso de los hiperparámetros para mejorar el desempeño de los modelos.

13 BALANCEO DE DATOS Y AFINAMIENTO DE HIPERPARÁMETROS

13.1 BALANCEO DE DATOS

Para poder obtener un Dataset con datos balanceados se aplicó el método SMOTE que genera datos sintéticos con el valor 0 para la variable cliente_aceptado.

13.2 AFINAMIENTO DE HIPERPARÁMETROS RANDOM FOREST CON SMOTE

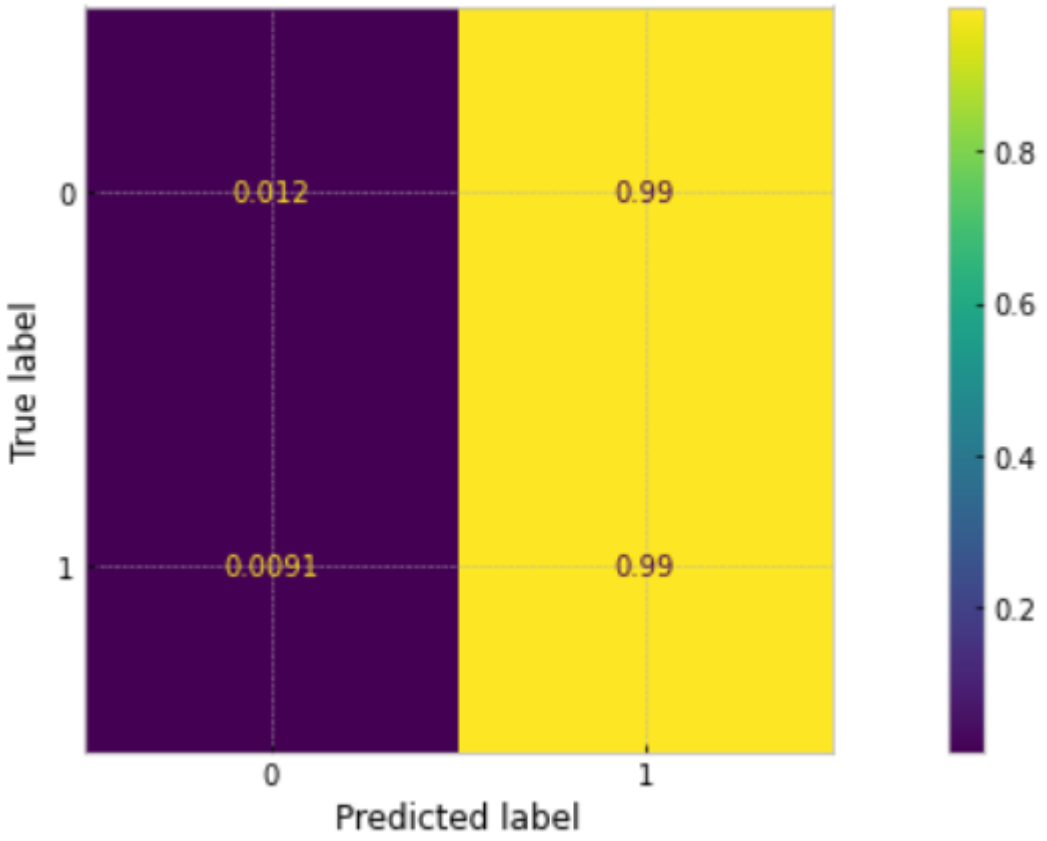
Se utilizó el método HalvingGridSearch para optimizar los hiperparámetros n_estimators, criterion, max_samples_split, max_samples_leaf y max_features, con la siguiente grilla:

```
params_grid = {'n_estimators': [512, 1024, 2048],
               'criterion':['entropy','gini'],
               'min_samples_split': [2,5,10,20],
               'min_samples_leaf': [4,5,6],
               'max_features': [0.1, 0.5, 1.0]
               }
```

Los mejores valores de los hiperparámetros mediante el afinamiento fueron los siguientes:

	criterion	max_features	min_samples_leaf	min_samples_split	n_estimators	mean_test_score
320	entropy	1.0	5	20	2048	0.914717

Las métricas obtenidas con el modelo entrenado con los hiperparámetros afinados fueron los siguientes:




```
precision 0.83
recall 0.99
f1_score 0.9
accuracy 0.82
```

Luego de balancear los datos mediante el método SMOTE y afinar los hierparámetros en el modelo de Random Forest, no se pueden observar mejoras significativas en los valores predichos.

13.3 AFINAMIENTO DE HIPERPARÁMETROS REGRESIÓN LOGÍSTICA

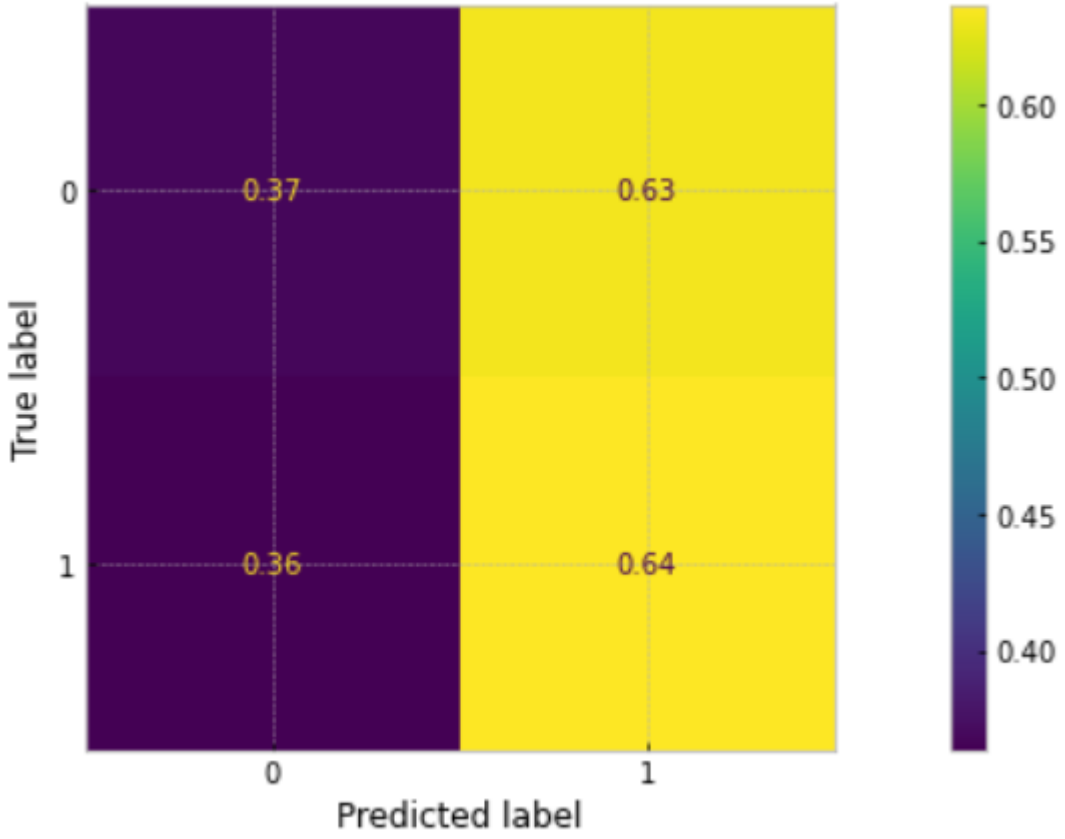
Se utilizó el método HalvingGridSearch para optimizar los hiperparámetros penalty y solver, con la siguiente grilla:

```
params_grids = {
                'penalty': ['l2'],
                'solver':['liblinear', 'newton-cg', 'lbfgs', 'sag', 'saga']
                }
```

Los mejores valores de los hiperparámetros mediante el afinamiento fueron los siguientes:

```
Mejores parametros {'penalty': 'l2', 'solver': 'newton-cg'}
```

Las métricas obtenidas con el modelo entrenado con los hiperparámetros afinados fueron los siguientes:



```
precision 0.83
recall 0.64
f1_score 0.72
accuracy 0.59
```

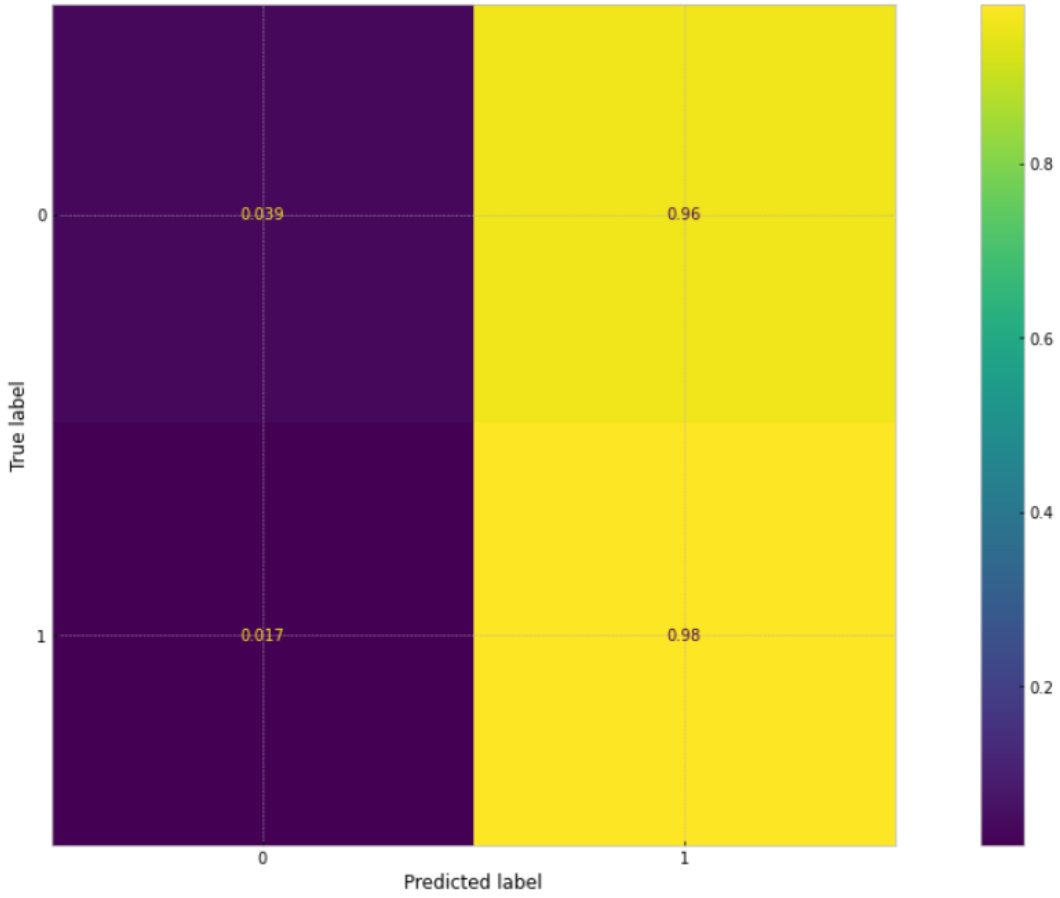
Antes de realizar el balanceo de datos el modelo predecía muchos falsos negativos, es decir rechazaba al 81% de los clientes.

Luego de balancear los datos mediante el método SMOTE y afinar los hierparámetros en el modelo de Regresión Logística, se puede observar que disminuyeron los falsos negativos a un 36%, y a su vez el modelo predice de manera correcta en un 64% a los clientes que deben ser aceptados y en un 37% a los clientes que deben ser rechazados.

13.4 MCA - ANÁLISIS DE CORRESPONDENCIAS MULTIPLES

Se utilizó el método de MCA para reducir la cantidad de variables categóricas con las que se entrenan los modelos.

Las métricas obtenidas luego de aplicar MCA en un modelo Random Forest son las siguientes:




```
precision 0.83
recall 0.98
f1_score 0.9
```

Luego de aplicar MCA los resultados obtenidos no mejoraron con respecto a los modelos entrenados anteriormente. Esto puede corresponder a que los coeficientes obtenidos con MCA acumulan aproximadamente el 28% de la varianza.

Finalmente, se concluye que no es conveniente utilizar este método.

14 CONCLUSIÓN

Para que los bancos puedan decidir de manera más eficiente que clientes deben ser aceptados y cuales rechazados a la hora de otorgarles un crédito, se recomienda utilizar un algoritmo de machine learning, ya que estos permiten optimizar los de decisiones y minimizar errores.

A lo largo del desarrollo del proyecto se entrenaron diversos modelos, para los cuales se balancearon los datos disponibles, mediante la técnica de SMOTE y se realizaron afinamientos de hiperparámetros.

El modelo que mayor desempeño obtuvo es el de Regresión Logística con el afinamiento de hiperparámetros realizados en donde se obtuvo una tasa de verdaderos positivos (cliente aceptado) del 64%, una tasa de verdaderos negativos (cliente rechazado) del 37% y un f1-score de 0.72. A pesar de que el f1-score del Random Forest es mayor, no se escogió este modelo ya que acepta a todos los clientes, por lo que es un mal modelo.

Finalmente, a pesar de que el uso del modelo de regresión logística puede agilizar el trabajo de los bancos los valores de las métricas resultantes no son suficientemente confiables, por lo que un analista debería verificar que los resultados arrojados por el modelo sean correctos. Para mejorar el modelo se recomienda recomendar más datos de clientes rechazados.