PREDICCIÓN DE APROBACIÓN DE TARJETAS DE CRÉDITO

Autores: Adrián Dondo, Patricio Novellino, Sebastián Pansecchi, Agustín Pretini

Comisión 29805

AGENDA

1- Contexto y Audiencia

2 - Hipótesis / Preguntas de Interés

3 - Overview de los datos

4- Análisis Exploratorio

5- Insights y Recomendaciones

6- Implementación de modelos:

- KNN

- Random Forest

7 - Conclusiones

Análisis final de las métricas e Insights

PREGUNTAS DE INTERÉS

Contexto

Las tablas de puntaje crediticio son un método común de control de riesgos en la industria financiera a la hora de mitigar el sesgo humano y homogeneizar las respuestas a millones de solicitudes de préstamos.
Lo que subyace en éste método, es un estudio de la probabilidad de que una futura operación de un importe determinado acabe entrando en mora. Si esa probabilidad es inferior al límite que designe la entidad financiera, la operación será viable, de lo contrario la solicitud saldrá rechazada.

Audiencia

Este análisis intentará ayudar a una entidad bancaria a la hora de decidir si emite una tarjeta de crédito al solicitante. Los puntajes de crédito pueden cuantificar objetivamente la magnitud del riesgo.
La entidad financiera estaría capacitada para tomar decisiones internas, conociendo el riesgo que le representaría cada cliente.

Limitaciones

Para éste análisis se utiliza una base de datos con una muestra (aunque bastante grande) limitada de observaciones. Además los datos son de una entidad financiera de Estados Unidos, por lo cual es posible que ciertas variables no sean del todo trasladables a un análisis en nuestro país.

PREGUNTAS DE INTERÉS

Preguntas Primarias - Objetivos

- ¿Cuáles son los buenos pagadores?
- ¿Cuáles son los malos pagadores?
- ¿Qué acciones podría emplear el banco?

Preguntas Secundarias

- ¿Qué tipo de empleo tienen las personas que más demoran en pagar? ¿Y los que menos demoran? ¿Y qué antigüedad tienen?
- ¿Cuál es el nivel de estudios que tienen las personas con más atraso en el pago?
- ¿Qué estado civil tienen las personas que más demoran en pagar?
- ¿Influye el nivel de ingresos en el atraso en el pago?

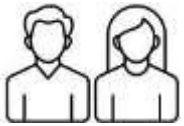
OVERVIEW DE LOS DATOS

775K
registros

5 categorías de
ingresos/ocupación

3 categorías de pago
(status)

5 niveles de
educación



5 categorías de
estado civil

ANÁLISIS EXPLORATORIO

Trabajamos las tablas para poder crear un modelo de clasificación que permita predecir la variable **'status'**, que refiere al nivel de atraso del cliente en pagar al banco.

Para poder trabajar de una mejor manera, agrupamos los días de retraso de pago en 3 categorías:

- 0: no tomadores de préstamos
- 1: buenos pagadores
- 2: malos pagadores

Las variables clave para intentar predecir **'STATUS'** serían:

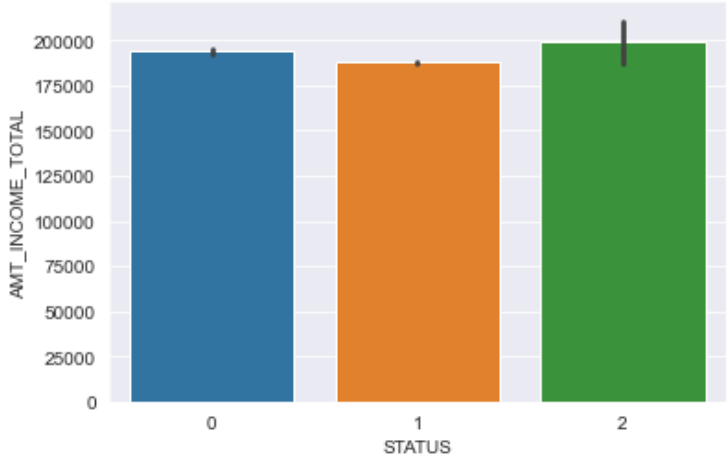
- **'AMT_INCOME_TOTAL'**(ingreso total)

- **'NAME_EDUCATION_TYPE'** (Nivel de educación)

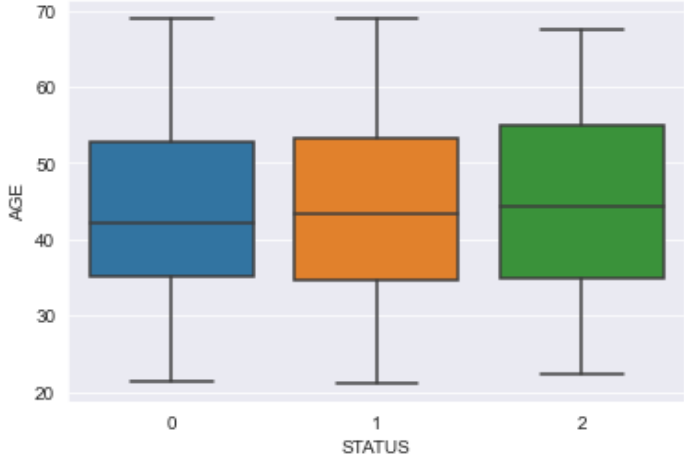
- **'OCCUPATION_TYPE'** (tipo de ocupación)

- **'NAME_FAMILY_STATUS'** (Estado civil), entre otras.

ANÁLISIS EXPLORATORIO



Observamos que el **salario promedio** es similar en las 3 categorías, con lo cual no podemos afirmar que haya una relación entre la demora del pago y el monto de ingresos.



La mayoría de las personas de cada grupo de status tienen **edades similares.** Esto es lógico, ya que se centra mayoritariamente en la edad laboral activa de las personas, de entre 30 y 55 años.

ANÁLISIS EXPLORATORIO

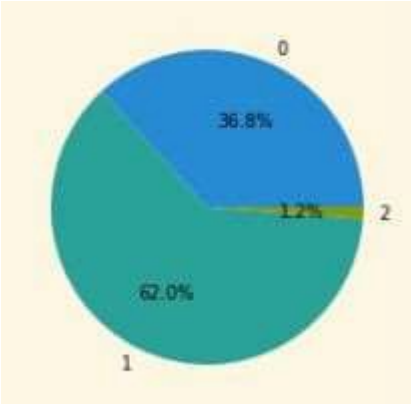
El barplot muestra como es la distribución de ingresos en las 3 categorías según el tipo/categoría de cliente.

Observamos que el comportamiento es similar en todas las categorías, con un **leve aumento** de "**commercial associate**" que son malos pagadores.

Como se observa en el gráfico de torta, la **mayoría** de los usuarios son **buenos pagadores,** lo cual indica que es negocio de bajo riesgo. Una gran parte no pidió préstamo alguno y la menor parte son malos pagadores (**sólo el 1,2%)**.

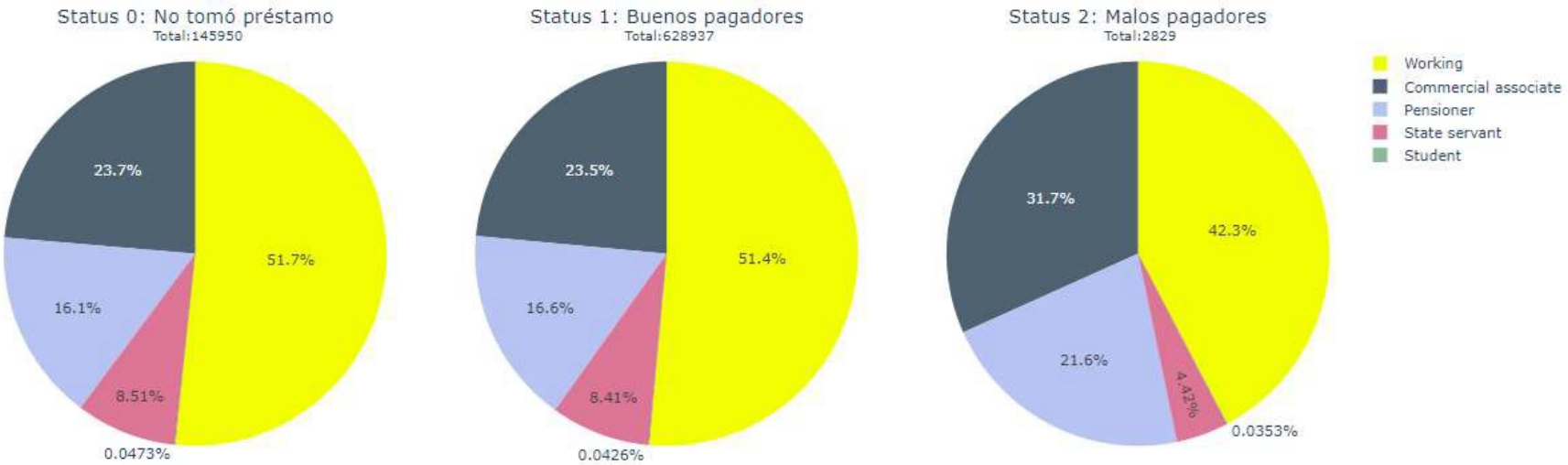
Con respecto a esto destacamos que al ser tan bajo el porcentaje de malos pagadores; hay una enorme oportunidad, que se podría explotar atrayendo al **37% de usuarios que no toma préstamos**.

Por otra parte, el hecho de que las clases estén tan desbalanceadas representa un desafío que debemos superar para poder entrenar nuestro modelo.



ANÁLISIS EXPLORATORIO

Categorías de clientes



Se muestra que proporción representa cada tipo de empleo dentro de cada nivel de status. La idea es ver si al pasar de buenos a malos pagadores hay alguna categoría de empleo que crezca o decrezca considerablemente. Inicialmente teníamos la hipótesis de que los "**comercial associate**" eran mejores pagadores que los "**working**", la cual se ve derribada a simple vista observando las tortas.

ANÁLISIS EXPLORATORIO

Del gráfico de categoría de clientes se desprenden las siguientes conclusiones:

- A medida que la categoría de status va creciendo se observa que los **commercial associate** y los **pensioner** van aumentando su participación, es decir, que tienen una mayor demora en los pagos.
- Los **state servants** y los **workings** van disminuyendo su participación a medida que aumenta la categoría.

Pareciera que no existe una relación entre el ingreso y el atraso en los pagos, ya que los **commercial associate** obtienen mayores ganancias que el resto, y son los que porcentualmente más crecen al comparar buenos con malos pagadores.

ANÁLISIS EXPLORATORIO

Nivel de estudios alcanzado

Se realiza un análisis similar que en la filmina anterior, pero segregando por nivel de estudios dentro de cada categoría de status.

ANÁLISIS EXPLORATORIO

Ratio de malos pagadores por nivel educativo alcanzado

```
Higher education: 0.4386026503396011 %
Secondary / secondary special: 0.32350298801551136 %
Incomplete higher: 0.4220383131656171 %
Lower secondary: 0.7741190063547082 %
Academic degree: 0.11933174224343675 %
```

Según el nivel de estudios alcanzados, podemos decir:

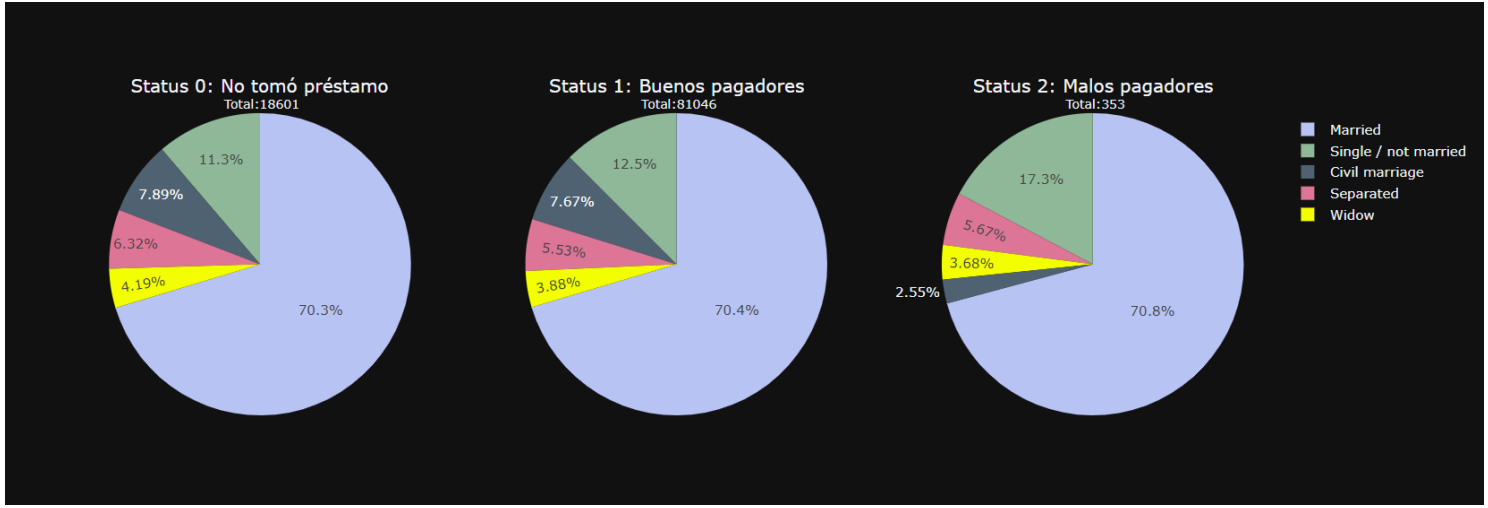
1. Para cada nivel de status, la distribución de las personas en función de su nivel educativo es muy similar en todos los niveles.

2. No obstante ello, al **incrementar el nivel de status**, se puede observar un **incremento porcentual** en la categoría **'Higher education'**, y un **decremento** en la categoría **'Secondary / Secondary special'**.

3. Además, detectamos que porcentualmente la categoría con mayor cantidad de **malos pagadores** es **'Lower secondary'**, aunque con un ratio no tan elevado (0,7%).

ANÁLISIS EXPLORATORIO

Estado Civil



Notamos prácticamente la **misma distribución** de usuarios según su **estado civil** dentro de cada categoría de status. No podemos concluir que las personas solas (que agrupan viudos, separados y solteros) tienen una mejor conducta crediticia que las personas en pareja (lo cual éra una hipótesis que teníamos).

INSIGHTS & RECOMENDACIONES

Propuesta 1

- Correr el ciclo de cierre de aquellos clientes catalogados como malos pagadores, de manera que el vencimiento del crédito sea posterior a la fecha de cobro de salarios.
- Así el cliente podría abonar en término y se lograría bajar el índice de mora.
- Se diseña una acción específica que se adapta a la necesidad del cliente, fidelizando a los mismos.

Propuesta 2

- Lograr que aquellos que no tienen préstamos (categoría 0) empiecen a tomarlos. Esto sería una enorme oportunidad a explotar por la entidad, ya que tal como se mencionó en la filmina 8, el porcentaje de malos pagadores es muy bajo en relación al total de clientes.
- ¿Cómo atraer a quiénes no toman préstamo? Remarcando los beneficios que trae al cliente.
- Aprovechar el historial crediticio del cliente, ofrecer préstamos preaprobados, premiar el consumo con puntos y premios, además de beneficios en Marketplace.

IMPLEMENTACIÓN DE MODELOS

Nuestra variable target es **'STATUS'**, la cual es categórica y posee 3 categorías. Por lo cual utilizaremos algoritmos supervisados de clasificación ya que se conoce a priori el número de clases. Los algoritmos a evaluar serán Random Forest y K Nearest Neighbor.

Para cada uno de estos, se implementó un entrenamiento del modelo base, luego empleamos una validación cruzada con StratifiedKFolds, para obtener las mejores particiones de train y test, y por último con el fin de optimizar, se entrenó un modelo con los mejores parámetros obtenidos con RandomizedSearchCV.

Adicionalmente, observamos que nuestra variable target ('STATUS') **posee desbalanceo de clases**, por lo cual entrenamos al modelo con los datos originales, pero al no ser los resultados satisfactorios, se empleó la técnica de oversampling 'SMOTETomek', para balancear las clases.

A continuación se realizará una breve descripción de ambos modelos, y luego pasaremos a detallar el proceso realizado para RandomForest, que fue el modelo que mejores resultados arrojó.

KNN

El algoritmo K Nearest Neighbors, se entrena para predecir la etiqueta de una observación nueva en base a la etiqueta de las k observaciones más cercanas a ella.

En nuestro caso comenzamos un modelo de las siguientes características:

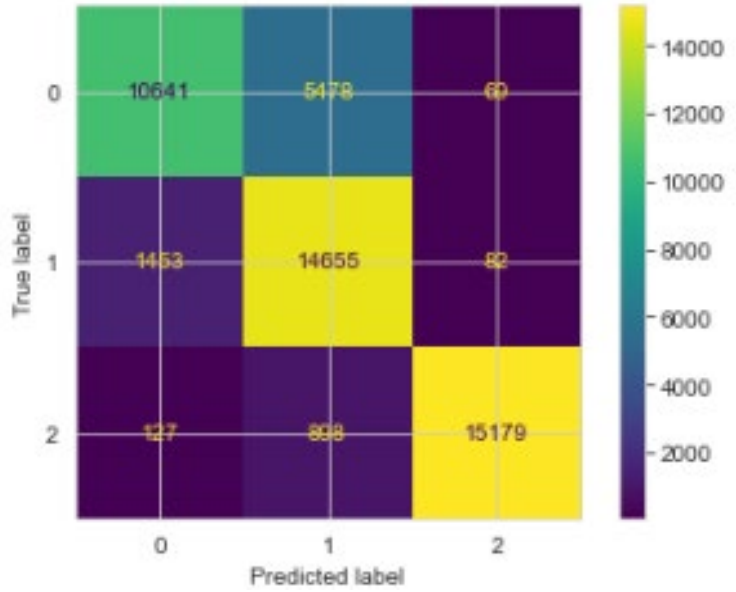
"KNeighborsClassifier(weights='distance', n_neighbors=7, p=2)"

Elegimos 7 vecinos porque es un número usado de manera stándard que suele dar buenos resultados, p= 2 para distancia euclideana y 'distance' para que pondere a los vecinos de manera inversa a su distancia respecto del punto observado.

Luego de aplicar StratifiedKFolds y RnadommizedSearchCV(skf), las mejores métricas las obtuvimos con el modelo base descripto, y tomando la mejor división de datos según skf. Las mismas se detallan a continuación:

KNN

	precision	recall	f1-score	support
0	0.87	0.66	0.75	16188
1	0.70	0.91	0.79	16190
2	0.99	0.94	0.96	16204
accuracy			0.83	48582
macro avg	0.85	0.83	0.83	48582
weighted avg	0.85	0.83	0.83	48582



RANDOM FOREST

Random Forest funciona creando un conjunto de árboles de decisión, donde cada árbol se entrena con un subconjunto aleatorio de datos y características. Luego, para hacer una predicción, los árboles realizan sus predicciones, las que se promedian o votan para obtener la predicción final.

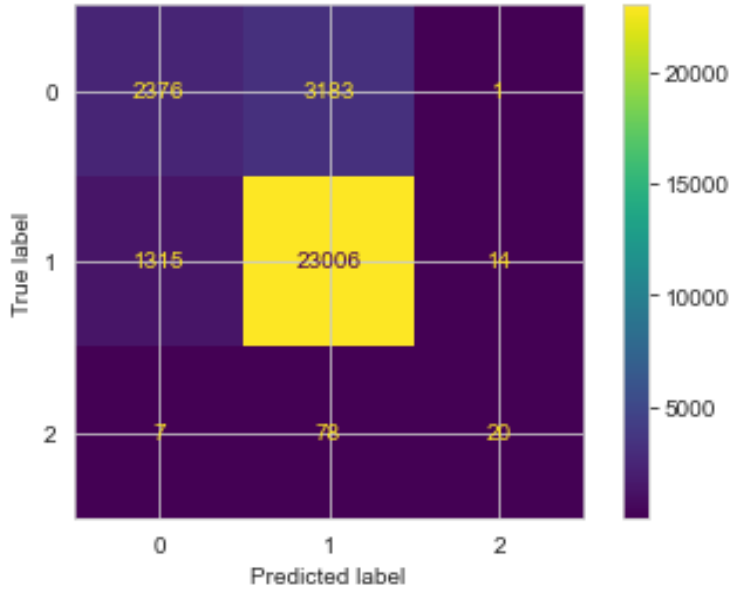
RANDOM FOREST

Comenzamos utilizando este modelo con los siguientes parámetros:

"RandomForestClassifier(random_state=42, n_estimators=20)"

El mismo arrojó los siguientes resultados:

	precision	recall	f1-score	support
0	0.64	0.43	0.51	5560
1	0.88	0.95	0.91	24335
2	0.57	0.19	0.29	105
accuracy			0.85	30000
macro avg	0.70	0.52	0.57	30000
weighted avg	0.83	0.85	0.83	30000



RANDOM FOREST

Las métricas a evaluar son las siguientes:
En cuanto al desempeño general, podemos observar que la clase 1 es explicada correctamente por el modelo. En específico una primera interpretación de los resultados obtenidos en esta métrica son:

• **Precisión:** Nos indica que la calidad del modelo es buena debido a que, en caso de aparecer "buenos clientes pagadores", los predecirá correctamente en la mayoría de los casos.

• **Recall:** Esta métrica nos da a entender con que facilidad el modelo identifica un cliente. En la clase mencionada (1) es elevada, ya que tenemos un número elevado de verdaderos positivos.

• **F1:** Nos indica que tenemos pocos falsos positivos y negativos, lo cual indica que tenemos altas probabilidades que un futuro cliente sea categorizado correctamente. Además esta métrica es muy buena para nuestro caso de estudio, ya que se presenta el desbalanceo de clases mencionado en el final.

•**Accuracy:** En general, las predicciones realizadas por el modelo se acercan a los datos reales; de todas maneras tuvimos cuidado con esta métrica ya que un valor muy cercano a 1 puede darnos a entender un caso de overfitting.
En contraposición, lo que encontramos fue que para el resto de las clases (0 y 2), los valores de las métricas se encontraban por la mitad o cercanas a 0. De esta manera se observa que el modelo no maneja bien estas clases para predecirlas y categorizarlas con facilidad y precisión, lo que reflejaría que el modelo se encuentra en un caso de underfitting (no podría predecir en base al score crediticio si es un mal pagador o un cliente que no tomó deuda este mes).

Para solucionar esto implementaremos la herramienta SMOTE-Tomek.

SMOTE-TOMEK

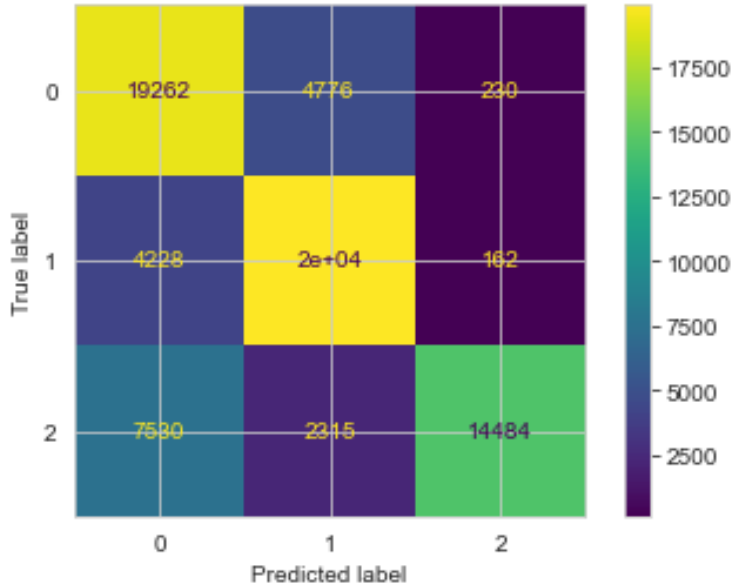
SMOTE-Tomek es una técnica de balanceo de clases que combina el uso de SMOTE (Synthetic Minority Over-sampling Technique) y Tomek Links. SMOTE se utiliza para generar datos sintéticos para las clases minoritarias en un conjunto de datos desbalanceado, mientras que Tomek Links se utiliza para eliminar aquellos puntos que están muy cerca de un punto de otra clase, lo que puede ayudar a mejorar la precisión del modelo y reducir el sobreajuste. SMOTE-Tomek se utiliza para tratar de equilibrar el conjunto de datos y mejorar el rendimiento del modelo de clasificación en casos en los que las clases son muy desbalanceadas.

Se procedió a aplicar esta técnica, y volver a entrenar el modelo.

SMOTE-TOMEK & RANDOM FOREST

Luego de realizar el procesamiento de SMOTE-Tomek, se obtuvieron las siguientes métricas:

	precision	recall	f1-score	support
0	0.62	0.79	0.70	24268
1	0.74	0.82	0.78	24272
2	0.97	0.60	0.74	24329
accuracy			0.74	72869
macro avg	0.78	0.74	0.74	72869
weighted avg	0.78	0.74	0.74	72869



SMOTE-TOMEK & RANDOM FOREST

Si bien en la clase 1, se vio empeorada la performance del modelo; obtenemos métricas más uniformes para todas las clases, con notables mejoras en clases o y 2. A pesar de esto, no consideramos suficiente el resultado obtenido. Por ello procedimos a aplicar una validación cruzada con **StratifiedKFolds** y de esta manera, nos quedamos con la partición de nuestros datos que mejores métricas arrojó.

STRATIFIED K FOLDS

StratifiedKFold es un método de validación cruzada que divide el conjunto de datos en subconjuntos de entrenamiento y prueba. La principal ventaja es que en lugar de una división aleatoria, mantiene la proporción de clases en cada uno de los subconjuntos. Esto es especialmente útil cuando tenemos un desbalance de clases en nuestros datos, ya que garantiza que cada subconjunto tenga una representación adecuada de cada clase.

Para utilizar StratifiedKFold, debemos especificar el número de "pliegues" o divisiones que queremos hacer. Por ejemplo, si establecemos n_splits=5, StratifiedKFold dividirá nuestro conjunto de datos en 5 subconjuntos y realizará una validación cruzada en cada uno de ellos. Esto significa que, en cada iteración, un subconjunto diferente se utilizará como datos de prueba y el resto se utilizará como datos de entrenamiento.

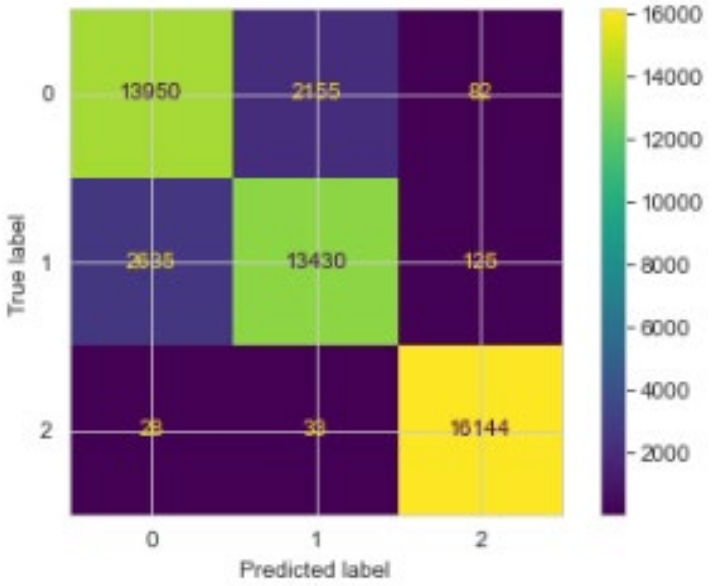
Nosotros utilizamos skf con estos parámetros:

"StratifiedKFold(n_splits=5, random_state=40, shuffle=True)"

STRATIFIED K FOLDS

Luego de aplicar la técnica descripta, y de haber seleccionado la mejor partición de nuestros datos las métricas que obtuvimos fueron las siguientes:

	precision	recall	f1-score	support
0	0.84	0.86	0.85	16187
1	0.86	0.83	0.84	16190
2	0.99	1.00	0.99	16205
accuracy			0.90	48582
macro avg	0.90	0.90	0.90	48582
weighted avg	0.90	0.90	0.90	48582



RANDOMIZEDSEARCHCV

Si bien las métricas obtenidas con el modelo anterior ya fueron muy buenas, procedimos a aplicar un RandomizedSearchCV, para probar diferentes combinaciones entre hiperparámetros, para intentar lograr aún una mejor performance. La grilla fue aplicada con los datos correspondientes a la mejor partición según skf.

Esto nos llevó a entrenar un modelo de las siguientes características:
"RandomForestClassifier(random_state=42, n_estimators=10, max_features= 'log2', max_depth=5,criterion= 'entropy')"

Y arrojó los siguientes resultados:

	precision	recall	f1-score	support
0	0.47	0.32	0.38	16187
1	0.53	0.47	0.50	16190
2	0.60	0.85	0.70	16205
accuracy			0.55	48582
macro avg	0.53	0.55	0.53	48582
weighted avg	0.53	0.55	0.53	48582

Tal como se puede ver, hubo una clara merma en las métricas, motivo por el cual decidimos quedarnos con el modelo base que habíamos implementado, con los datos luego del oversampling y seleccionando la mejor partición según skf.

CONCLUSIONES

Podemos ver que el modelo que mejores resultados arrojó fue un random forest de las siguientes características: **"RandomForestClassifier(random_state=42, n_estimators=20)"**, utilizando la mejor partición de datos train/test según el méto do StratifiedKFolds, y la técnica de oversampling SMOTETomek.
El mismo arrojó estos resultados:

	precision	recall	f1-score	support
0	0.84	0.86	0.85	16187
1	0.86	0.83	0.84	16190
2	0.99	1.00	0.99	16205
accuracy			0.90	48582
macro avg	0.90	0.90	0.90	48582
weighted avg	0.90	0.90	0.90	48582

Por esta razón, decidimos recomendar al banco el modelo de las características mencionadas.

Tiene sentido que de mejor el RandomForest por sobre knn; ya que RandomForest implementa más de un algoritmo (árbol) ens imultáneo, lo cual lo vuelve más preciso y resistente al overfitting.

CONCLUSIONES

Análisis final de las métricas e insights

Los resultados del modelo muestran que el modelo tiene una precisión y un recall bastante altos. A continuación, se explicarán cada una de las métricas:

- **Precision**: La precisión es la fracción de predicciones positivas que son correctas. En nuestro caso, la precisión para las clases **0, 1 y 2** es de **0.84, 0.86 y 0.99** respectivamente. Esto significa que el modelo es capaz de predecir correctamente el 84% de los clientes que no tomaron préstamos, el 86% de los buenos pagadores y el 99% de los malos pagadores.

- **Recall**: El recall es la fracción de verdaderos positivos que son detectados por el modelo. En este caso, el recall para las clase s **0, 1 y 2** es de **0.86, 0.83 y 1.00** respectivamente. Esto significa que el modelo es capaz de detectar el 86% de los clientes qu e no tomaron préstamos, el 83% de los buenos pagadores y el 100% de los malos pagadores.

- **F1score**: El f1score es una métrica que combina precision y recall y se calcula como la media armónica de ambas. En nuestro caso, el f1 score para las clases **0, 1 y 2** es de **0.85, 0.84 y 0.99** respectivamente. Esto significa que el modelo tiene una prec isión y un recall bastante similares para las clases 0 y 1, y una precisión y un recall muy altos para la clase 2.

CONCLUSIONES

Teniendo esto en cuenta, queremos destacar que el modelo tiene una performance prácticamente perfecta en la clase 2 (Mal ospagadores); por lo cual si un cliente es clasificado como mal pagador, recomendamos a la empresa no otorgar el préstamo.

Mientras tanto, en las clases 0 (No toma préstamo) y 1 (Buen pagador), si bien el rendimiento del modelo es muy bueno, pre senta un pequeño margen de error. Para solucionar esto, sugerimos realizar en estos un análisis mas pormenorizado para cada cliente clasificado en clases 0 y 1 antes de tomar una decisión al respecto.