Práctica de Aprendizaje Autónomo
Predicción de cierre de cuenta bancaria

Hugo Aranda Sánchez
Albert Ruiz Vives
Departamento de Computación

10/01/2024

Índice

1 Introducción

Los humanos somos impredecibles en general, pero a través de la conducta y las observaciones podemos predecir un poco como actuaremos en el futuro. Esto es un gran recurso a explorar para las empresas que ofrecen servicios, pues les interesa al máximo conocer y entender como los usuarios están interactuando con su servicio para mejorarlo y mantenerse competitivos y relevantes.

En este marco se encuentra nuestro proyecto el cual hace uso de unos datos extraídos de de kaggle llamado Credit Card customers. El cual como su propio nombre indica contiene información relacionada con los clientes de un banco teniendo múltiples valores relacionados con su actividad en el mismo y el uso de sus tarjetas de crédito.

Hemos seleccionado esta temática pues nos parece un proyecto de un gran valor para las empresas. Pues saber predecir este efecto tiene un gran impacto en el porcentaje de clientes se van si se toma acción. Al usar este estilo de predicción las empresas pueden lanzar campañas de fidelidad que mejoren enormemente los ratios de clientes que se quedan y consigan tener un gran impacto en el bienestar y funcionamiento de la empresa.

El conjunto de datos en particular sobre el que vamos a trabajar ha sido extraído de la pagina web de kaggle. Este es el Credit Card Customers [5], el cual es dominio público y hemos extraído por tener una cantidad muy alta de muestras la cual nos permitirá alimentar nuestro modelo sea cual sea este.

2 Objetivos del Proyecto

Por lo tanto, al estar enfrentándonos frente a un problema de clasificación nuestro principal objetivo es el entrenamiento de modelos que sean capaces de clasificar a los clientes según si este abandonara o no el servicio.

Por ello, intentaremos predecir nuestra variable objetivo *Attrition_Flag* la cual nos indica si el usuario ha cerrado su tarjeta o no. De esta manera pudiendo detectar a tiempo aquellos clientes que quieran rescindir su servicio de forma inminente.

Adicionalmente en el marco académico pretendemos poder aplicar todos los conocimientos obtenidos con un problema real para poner en uso todas las técnicas trabajadas durante la asignatura.

Además de la propia predicción planeamos obtener un conocimiento más profundo de nuestro dataset que nos permita no solo trabajar mejor con el mismo a la hora de confeccionar modelos sino poder comprenderlo mejor por si mismo para beneficio de la empresa (entender diversos patrones de comportamiento y que elementos se relacionan).

3 Datos disponibles

Este es un problema popular dentro del Machine Learning así que podemos obtener datos relevantes sobre el estado de este problema en el mundo real a través de diferentes proyectos anteriores.

De estos principalmente podemos aprender como es común valorar los resultados de nuestros experimentes en el sector. Por ejemplo, en un artículo que trata sobre el tema [1] podemos observar cómo para un dataset relativamente similar este naturalmente tiene un especial cuidado y atención por los falsos negativos pues desde el punto de vista empresarial estos son críticos. Esta idea formara parte de nuestro razonamiento para valorar los modelos de entre otros aspectos. También podemos ver que aspectos son importantes en general para representar la eficiencia en papers de este estilo.

También encontramos un artículo el cual contiene una valoración interesante del proyecto la cual nos permite afilar un poco que modelos sentimos interés por probar como por ejemplo modelos de Gradient Boosting [2], Suport Vector Machines [4] y Bosques Aleatorios [3]. Esto nos hace pensar que la no linealidad de los modelos es vital. Pues como mencionamos anteriormente, en un conjunto de datos como este puede haber muchos tipos diferentes de situaciones los cuales generan patrones complejos.

4 Proceso de exploración de los datos

Para explorar nuestros datos hemos decidido visualizar y explorar las variables individuales y la correlación entre las mismas. Así como métodos de visualización más complejos que nos permiten observar patrones ocultos en los datos y entender mejor su comportamiento.

4.1 Partición de datos

Primero de todo y antes de poder realizar cualquier visualización separamos entre datos de entrenamiento y prueba. Pues no podemos visualizar el conjunto entero ya que esto nos daría información sobre el conjunto de prueba el cuál para nosotros debe actuar como una caja negra con la que realizamos dichas pruebas. Esta partición es de 80/20.

4.2 Visualización Simple

Primero de todo para el proceso de exploración de los datos y antes del preproceso realizamos una tarea importante de visualización para obtener información sobre cómo se distribuyen nuestros datos y obtener posibles conocimientos sobre los mismos de cara a la selección de características y el preproceso.

Lo primero de todo observemos la distribución dentro de nuestra variable objetivo con un gráfico1 para ver a que nos enfrentamos. En este caso vemos que el porcentaje de las personas que se van esta entorno a un 17% el cual asienta uno de los principales factores a tener en cuenta durante el estudio que es el balanceo de clases. Que las clases estén desbalanceadas tiene implicaciones tanto en la eficiencia de los modelos en si como en la veracidad de los métodos que usamos para asignarles sus parámetros.

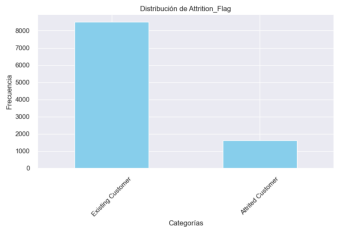


Figure 1: Visualización de los clientes que huyen

A nivel estadístico podemos observamos que existen tres rangos distintos entre los valores. Pues hay algunos valores referentes al límite del crédito o al total de dinero en transacciones que es muy alto y luego encontramos otros cómo conteos de interacciones que tienen un rango intermedio así como valores que oscilan entre un rango mínimo cómo el número de relaciones o personas dependientes. Esto nos da la pista de que tal vez debamos estandarizar para evitar que estas diferencias de magnitud pesen en la eficiencia de nuestros modelos.

Adicionalmente observamos mediante boxplots que en las variables que tratan sobretodo en cantidades de dinero tenemos presentes outliers que si bien tienen mucho sentido (se corresponden a aquellos clientes más pudientes) pueden hacer que nuestros modelos se desequilibren un poco.

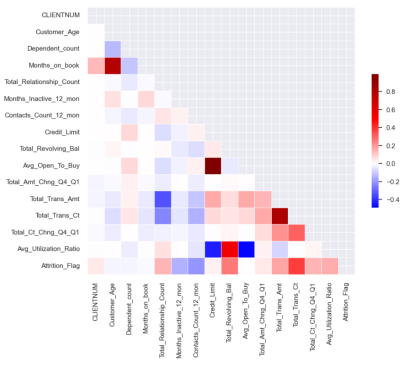


Figure 2: Visualización de las correlaciones

Después en las correlaciones observamos que attrition flag esta directamente relacionado con cualquier valor que implique actividad dentro del banco, en especial aquellas relacionadas con las transacciones y el saldo (y no tanto de otras como el crédito). Y esta inversamente relacionado con la inactividad. No se relaciona tanto con aspectos del estilo de vida del usuario cosa que tomaremos en cuenta a la hora de hacer descartes.

Estas variables que presentan gran correlación con la variable objetivo podemos ver si separamos los gráficos partiendo por el Attrition Flag el grupo que se va del grupo que se queda muestran distribuciones completamente diferentes3. Cosa que nos hace ver que a través de estos valores contienen dos tendencias diferentes. En otros casos como la edad parece no haber una gran correlación pues siguen una distribución normal similar a la que siguen las columnas naranjas.

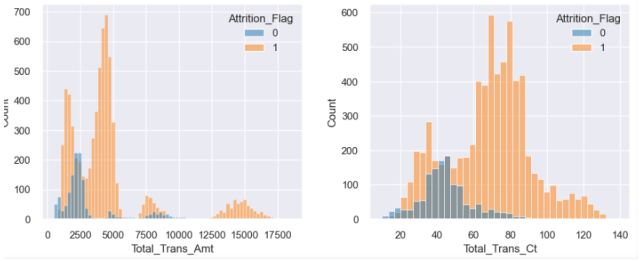


Figure 3: Gráficas de Total_Trans_Ampt y Total_Trans_Ct

4.3 Selección de características

De todas las variables las que seleccionaremos (a parte de Attrition_Flag la cual es nuestra variable objetivo) son:

- Variables numéricas
  - *Total_Relationship_Count*: Número de productos propiedad del cliente
  - *Months_Inactive_12_mon*: Meses inactivo en los últimos 12 meses
  - *Contacts_Count_12_mon*: Número de contratos en los últimos 12 meses
  - *Total_Revolving_Bal*: El saldo rotativo del cliente (cupo de crédito personal reutilizable)
  - *Total_Amt_Chng_Q4_Q1*: El cambio en el volumen de las transacciones
  - *Total_Trans_Amt*: El volumen de transacciones
  - *Total_Trans_Ct*: La cantidad de transacciones (conteo)
  - *Total_Ct_Chng_Q4_Q1*: El cambio en el conteo de transacciones
  - *Avg_Utilization_Ratio*: El ratio de utilización
- Variables categóricas
  - *Gender*: género del propietario de la cuenta
  - *Education_Level*: Nivel de educación del cliente
  - *Marital_Status*: Estado marital del cliente
  - *Income_Category*: Categoría de ingresos del cliente
  - *Card_Category*: Categoría de tarjeta de crédito (Blue, Silver, Gold, Platinum)

Las variables númericas parecían todas interesantes pero al investigar las correlaciones podemos destacar que de entre ellas hay cinco que no aportan correlación en los apartados que consideramos más interesantes. Estas son las relacionadas con el estilo de vida Customer_age y Dependent_Count las cuales no parecen significar mucho dentro del gran esquema de las cosas, pues parece más relevante las condiciones económicas que el resto de factores. En lo que respecta a Months_on_book, Credit_Limit y Avg_Open_To_Buy a pesar de estar relacionadas no parecen pesar lo suficiente.

4.4 Preproceso

Los principales retos a partir del preprocessing son la presencia de unas variables categoricas a ser recodificadas, las cuales fueron estudiadas y recodificadas siguiendo una cierta lógica. Para codificar tuvimos en cuenta que hay variables categóricas (la mayoría) que presentan un cierto orden en las categorías que representan. Cómo por ejemplo el nivel de educación (de bajo a alto) o los tramos de nivel de sueldo (de menos a mas). Por lo tanto mapeamos esto valores según esta progresión con enteros.

Adicionalmente, realizamos estandarización de variables la cual sacamos a relucir mediante la visualización de ciertos outliers.

Además la codificación de las variables categóricas introduce un problema que antes no teníamos. Muchas categóricas presentan una categoría *Unkown* que hemos aprovechado para poner como *missing values* y poder imputarlos mediante el `KNNImputer`. Este se encarga de imputarlo mediante el uso de los k vecinos más cercanos. Método no lineal adecuado para el problema que trabajamos.

4.5 Visualización Avanzada

Tras la visualización mediante métodos avanzados de nuestros datos obtuvimos los siguientes resultados.

4.5.1 PCA

El Análisis de Componentes Principales (PCA) a pesar de su potencial de reducción de dimensionalidad no llegamos a obtener una visualización muy esclarecedora.

Nuestro problema necesita una dimensionalidad relativamente alta para explicar toda su varianza pues para un 90% de varianza explicada nos encontramos que necesitamos 11 componentes principales. Adicionalmente los componentes al observar las variables que los componen observamos que el primer componente contiene variables que presentaban una alta correlación con la variable objetivo. Desgraciadamente PC2 y PC3 tienen variables que no nos interesan del todo aportando linealmente así que esas dimensiones no nos muestran mucho.

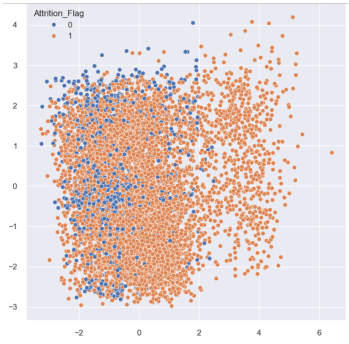


Figure 4: Visualización 2D de PCA

Podemos ver claramente en el gráfico que a lo largo del eje de PC1 podemos más o menos clasificar pero no se observan clusters claros de fugitivos. Existen zonas seguras dónde todos se quedan pero hacia los valores negativos de PC1 hay incerteza. Esto es potencialmente por las limitaciones de PCA el cual principalmente trata de observar correlaciones lineales y por lo tanto no puede identificar relaciones no lineales o patrones con fronteras difusas.

4.5.2 No lineales

Por lo tanto, recurriendo a métodos no lineales podemos mirar de capturar esas relaciones no lineales que andamos buscando y que por lo que hemos visto en la exploración de otros proyectos marca esta clase de problemas.

En esta categoría encontramos T-SNE. Este es capaz de capturar estructuras más complejas y funciona mapeando los datos de alta dimensionalidad en un espacio 2D y 3D (probaremos ambos). De esta manera puntos cercanos en espacios de alta dimensionalidad lo estarán en este nuevo espacio.

Como podemos apreciar este realiza un mejor trabajo (nuestra mejor versión es la que parte de los datos de PCA y luego aplica T-SNE). Ahora vemos un cluster en el centro de clientes que abandonan el servicio y aunque sigue habiendo ruido promete más.

Probamos también con isomap pero este no dio unos resultados del todo satisfactorios. Se agrupaba de formas que no nos aportaban mucho.

Figure 5: Visualización 2D de T-SNE partiendo de PCA

4.6 Protocolo de Remuestreo

Como hemos mencionado anteriormente en la partición de los datos al trabajar con un conjunto de datos desequilibrado, donde una clase de clasificación representa un porcentaje significativamente menor que otra, es fundamental utilizar estrategias de validación cruzada que preserven la proporción de clases en cada pliegue.

Por eso mismo usaremos Sratified K Fold el cual realizara esa partición preservando la proporción de clases. Además de esta manera podemos evitar sobreestimar o subestimar ciertos aspectos pues si las proporciones son tan justas un mal corte puede hacer quedar muy mal a nuestro modelo cuando realmente no tiene problemas en particiones mejor representadas.

Como métodos de optimización de hiperparametros utilizaremos GridSeach y BayesSearch. El primero explora exhaustivamente el espacio de hiperparametros pero es algo computacionalmente costoso ya que evalua cada combinación pase lo que pase. Bayes Seach como su nombre nos da a intuir hace una búsqueda más inteligente y eficiente al considerar los resultados anteriores para determinar las siguentes combinaciones a probar enfocandose a lo que promete más. Usaremos el más exhaustivo hasta que los modelos sean demasiado costosos.

Escogemos 5 folds pues ya son suficientes para poder probar diferentes configuraciones. Shuffle se queda activado para mezclar los datos antes de dividirlos de manera que evitemos patrones específicos de la secuencia de los datos. Como número de iteraciones para los protocolos que lo requieren proponemos 5.

5 Resultados métodos Lineales

Comenzaremos por los resultados de los métodos lineales. Pues aunque estos parezca que no van a prometer mucho por lo que hemos descubierto durante la primera fase de visualización es interesante empezar por los métodos más fáciles por si encontramos por poco costo computacional un resultado decente o muy bueno de primeras.

5.1 Naïve Bayes

El primer modelo cuyos resultados comentaremos será el de Naïve Bayes. Este modelo es algo simple pues asume independencia condicional entre las características de las clases. Además de asumir una distribución normal en los datos, la cual hemos visto que tampoco siguen rigurosamente nuestros datos. Partiendo de estas suposiciones lo que hace es calcular la probabilidad de pertenencia a una clase dado un conjunto de características asignando a cada una de estas un valor de probabilidad.

Para indicar si el usuario se queda o se va vemos que usa en gran medida para definir esas probabilidades Total_Trans_Ct cosa que tiene sentido pues tiene una alta correlación con la variable objetivo.

Reporte de Clasificación

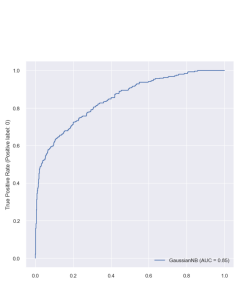
	precision	recall	f1-score	support
Attrited Customer	0.59	0.59	0.59	324
Existing Customer	0.92	0.92	0.92	1702
accuracy			0.87	2026
macro avg	0.75	0.76	0.76	2026
weighted avg	0.87	0.87	0.87	2026

Del reporte de clasíficación podemos observar que este se comporta relativamente bien para clasificar a los clientes existentes con un 92% de f1 score (la cual es una media ponderada de la precision y de la recogida (recall) pero falla muy a menudo con los usuarios que se van a ir.

Esto tiene sentido teniendo en cuenta que no es capaz de ver las relaciones no lineales que podíamos observar con el T-SNE y entonces cuando llega a la nube de puntos que contiene tanto usuarios que se quedan como los que se van es capaz de discernir entre el grupo más numeroso con facilidad.

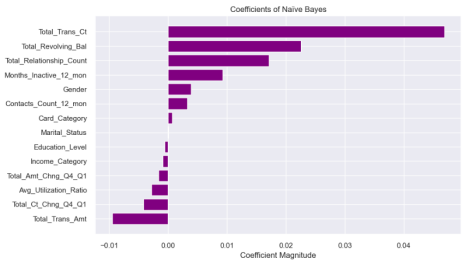
Aún así el valor de un 59% es aceptable para lo sencillo que es el modelo y nos pone un inicio prometedor de estudio con mucho margen de mejora.

Viendo el acierto general obtenemos un generoso 87% ya que la clase de existentes es muy numerosa y una buena puntuación ahí pesa mucho en el acierto general.



(a) GNB Roc Curve

(b) GNB Confusion Matrix



(c) Importancia de Componentes

Figure 6: Grafos de GNB

Si observamos la parrilla de gráficos podemos ver que la curva de Roc no se ciñe tanto como nos gustaria para los valores positivos pues en estos falla bastante.

A continuación, en la matriz de confusión podemos ver que falla entorno a 130 en falsos positivos y falsos negativos y solo estamos recolectando 191 de los clientes que se van.

Si a las asunciones que hace Naïve Bayes de normal sumamos el desbalance de clases y las extrañas fronteras no lineales es entendible sus ineficiencias pero supone una base sencilla y valiosa para iniciar el proyecto.

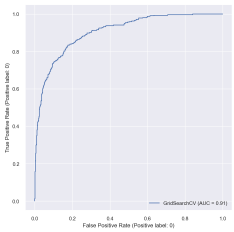
5.2 Logistic Regression

El siguiente método que hemos empleado es Logistic Regression. Éste es un método lineal al igual que el anterior, pero tiene ciertas ventajas respecto NB. Para empezar, es un modelo diseñado específicamente para trabajar con problemas de clasificación binaria (como es nuestro caso donde queremos saber si un usuario se marchará del banco o no). Además, este modelo no asume que las variables sean necesariamente totalmente independientes como NB. Por lo tanto podría mostrar un mejor rendimiento.

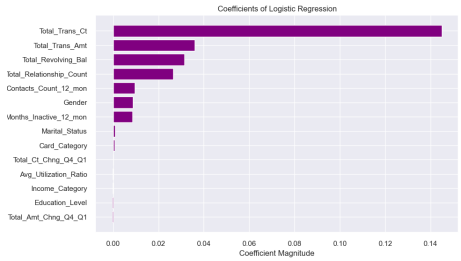
Reporte de Clasificación

	precision	recall	f1-score	support
Attrited Customer	0.48	0.79	0.60	200
Existing Customer	0.97	0.91	0.94	1826
accuracy			0.90	2026
macro avg	0.73	0.85	0.77	2026
weighted avg	0.93	0.90	0.91	2026

En el reporte de clasificación vemos que la "accuracy" general del modelo ha subido un poco respecto al anterior. Sobretodo debido a un mayor recall de los "Attrited Customers" (usuarios que marchan). No obstante, la precision de esta misma clase ha bajado considerablemente, resultando en una mayor tasa de falsos positivos como vemos en la confusion matrix a continuación:



(a) Logistic Regression Roc Curve



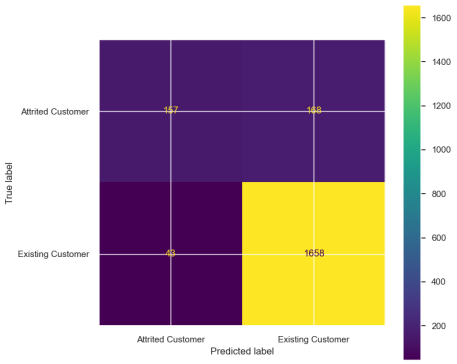
(b) Importancia de Componentes

Respecto a los componentes importantes, vemos que Total_Trans_Ct vuelve a destacar como el componente que más influye en la capacidad de decisión del diseño. Además, la curva ROC ha mejorado bastante, acercandose mucho más a la esquina superior izquierda (demostrando una mayor confianza en la predicción por parte del modelo).

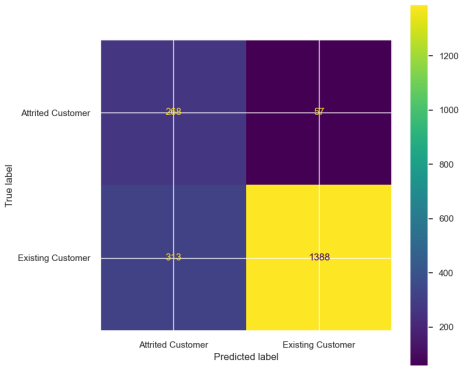
Como Logistic Regression intenta separar los individuos en las 2 clases mediante una función sigmoide, el hecho que el dataset esté desbalanceado puede afectar bastante en la predicción. Por ello hemos ajustado un modelo de Balanced Logistic Regression con estos resultados:

Reporte de Clasificación

	precision	recall	f1-score	support
Attrited Customer	0.45	0.84	0.59	325
Existing Customer	0.96	0.80	0.88	1701
accuracy			0.81	2026
macro avg	0.71	0.82	0.73	2026
weighted avg	0.88	0.81	0.83	2026



(a) Logistic Regression Confusion Matrix



(b) Balanced Logistic Regression Confusion Matrix

Figure 8: Comparación de las Confusion Matrix entre LR y Balanced LR

Como podemos ver, el Balanced Logistic Regression por lo general muestra peores resultados que los modelos anteriores con una accuracy general de un 81% (un 10% inferior que LR normal). Cabe destacar que aumenta mucho el recall de la clase que nos interesa, Attrited Customers. Sin embargo, esto lo hace a gran costa de precisión, cosa que empeora el rendimiento global del modelo. Podemos ver en la comparativa de confusion matrix que entre Logistic Regression y Balanced Logistic Regression estamos, a efectos prácticos, escogiendo entre tener una mayor tasa de Falsos Negativos (en el caso de LR) o Falsos Positivos (como es el caso de Balanced LR). Este compromiso es lógico ya que como método lineal, al equilibrar las clases lo que estamos haciendo es "mover" el punto de corte. En este caso consideramos preferible LR a Balanced LR por su mayor accuracy en general.

5.3 SVM Lineales & Cuadráticos

Los SVMs son modelos muy versátiles que se pueden aplicar a una gran extensión de problemas. En este apartado veremos 2 en concreto: SVM Lineales y Cuadráticos. Los SVMs lineales, tal como dice su nombre, mira de dividir el dataset en 2 clases haciendo uso de una línea, plano o hiperplano. Por lo tanto, al igual que con logistic regression, es útil observar los resultados por la simplicidad del modelo, aunque no esperemos los mejores resultados. Por otro lado, los SVM cuadráticos mapean los features del dataset a un plano superior para luego hacer la división ahí. Esto permite detectar relaciones no lineales entre los datos, cosa que podría ayudar bastante en este problema particular.

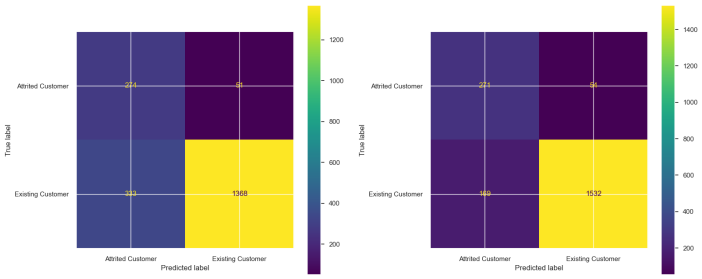
Reporte de Clasificación SVM Lineales

	precision	recall	f1-score	support
Attrited Customer	0.45	0.84	0.59	325
Existing Customer	0.96	0.80	0.88	1701
accuracy			0.81	2026
macro avg	0.71	0.82	0.73	2026
weighted avg	0.88	0.81	0.83	2026

Reporte de Clasificación SVM Cuadráticos

	precision	recall	f1-score	support
Attrited Customer	0.62	0.83	0.71	325
Existing Customer	0.97	0.90	0.93	1701
accuracy			0.89	2026
macro avg	0.79	0.87	0.82	2026
weighted avg	0.91	0.89	0.90	2026

Viendo los resultados de la clasificación, observamos como las SVM lineales tienen un rendimiento relativamente similar al Balanced Logistic Regresion, teniendo un buen f1-score para tanto la accuracy global como la clasificación de Existing Customers y teniendo buen recall para los Attrited Customers pero con una precision subpar. Por otro lado, los SVM Cuadráticos sí que presentan resultados dignos de comentar, obteniendo una f1-score de los Attrited Customers del 71%, el más alto obtenido hasta el momento. Este f1-score viene dado por un buen recall sumado a una precisión del 62% que deja mucho margen para mejoría pero ya empieza a ser un resultado mínimamente decente. Además, vemos que la accuracy total, con un f1-score del 89% también es bastante deseable.



(a) SVM Lineal Confusion Matrix (b) SVM Cuadrático Confusion Matrix

Figure 9: Comparación de las Confusion Matrix entre SVM lineal y polinómico

Esto mismo lo vemos si comparamos las confusion matrix de los dos modelos en la figura 9. Observamos que las SVM lineales tienen un altísimo número de falsos positivos, mientras que las SVM cuadráticas mantienen tanto los falsos positivos como los falsos negativos en valores considerablemente bajos. Esto nos parece confirmar la intuición que, tal como hemos visto en la visualización, modelos no lineales o que sean capaces de detectar este tipo de relaciones tendrán un mejor rendimiento.

6 Resultados métodos No Lineales

6.1 RBF SVM

Siguiendo con el uso de SVM para nuestra clasificación, la progresión natural son los RBF SVM. Este modelo también está basado en Support Vector Machines, su característica particular es que usa un kernel RBF también denominado Gaussiano en vez de uno polinómico como lo hacen las SVM Cuadráticas. Este kernel aplica una función matemática que mide las similitudes entre dos individuos, cosa que le permite encontrar de una forma efectiva las relaciones no-lineales entre datos.

Reporte de Clasificación

	precision	recall	f1-score	support
Attrited Customer	0.68	0.73	0.71	325
Existing Customer	0.95	0.94	0.94	1701
accuracy			0.90	2026
macro avg	0.82	0.83	0.82	2026
weighted avg	0.91	0.90	0.90	2026

Estos resultados ya empiezan a ser muy favorables. Es relativamente esperable ya que RBF SVM es una técnica muy popular para datasets con relaciones de datos no lineales, sin embargo, vemos otra vez una mejoría de Attrited Customers, el cual tiene el precision y recall mucho más equilibrados. Además Existing Customers presenta un mejor f1-score también. Esto hace que el modelo tenga una accuracy global de 90%, volviéndose el modelo más deseable visto hasta el momento. La Confusion Matrix y tabla de coeficientes de relevancia de features esta vez no presentan ninguna conclusión relevante interesante respecto a los otros modelos SVM.

6.2 Exploración con Decision Trees

Los siguientes modelos (Gradient Boosting y Random Forest) usan los Decision Trees como bloques de su fundación. Gradient Boosting aplica weak-learners (como serian los DT) iterativamente y Random Forest crea un "bosque" de éstos mismos. Por lo tanto, hemos querido hacer una exploración de si el oversampling o undersampling afectaba a los DTs en si para ver si uno es preferible al otro. Los resultados obtenidos son los siguientes:

En este grafo observamos que oversampling y undersampling parecen rendir muy ligeramente mejor, puesto que tienen una mayor area bajo la curva y además presentan una forma de curva que se acerca más a la esquina superior izquierda (especialmente undersampling). Por eso mismo hemos optado por aplicar undersampling a estos modelos con resultados bastante mejores.

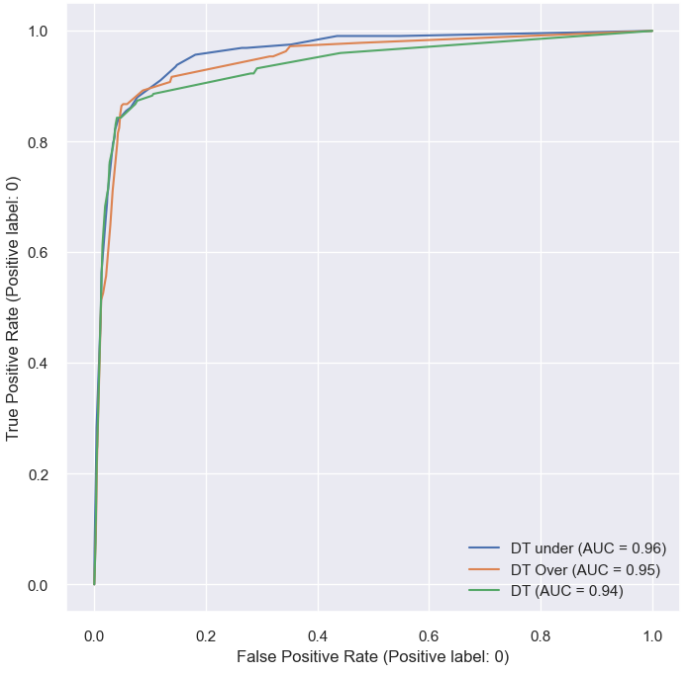


Figure 10: Curvas ROC de los DTs con diferentes métodos de sampling

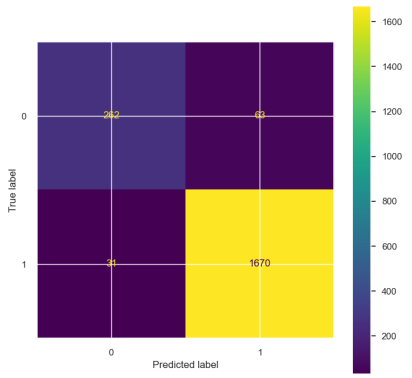
6.3 Gradient Boosting

Gradient Boosting es una técnica que consiste en aplicar sucesivamente weak-learners, con cada iteración aprendiendo de los errores de las anteriores iteraciones. Esto le permite ajustar muy bien las fronteras a la hora de clasificar hacia una clase u otra (por ejemplo usando DTs como weak-learners, esta técnica nos permite perfilar muy bien los varemos que decidirán si el modelo escoge una rama u otra para cada individuo). Los resultados obtenidos son los siguientes:

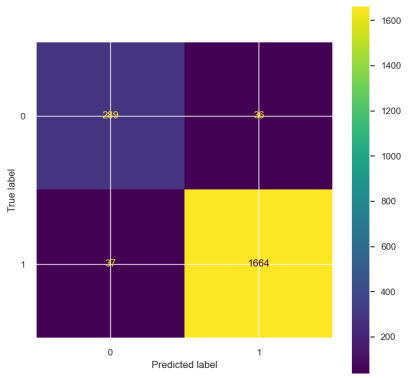
Reporte de Clasificación

	precision	recall	f1-score	support
Attrited Customer	0.89	0.89	0.89	325
Existing Customer	0.98	0.98	0.98	1701
accuracy			0.96	2026
macro avg	0.93	0.93	0.93	2026
weighted avg	0.96	0.96	0.96	2026

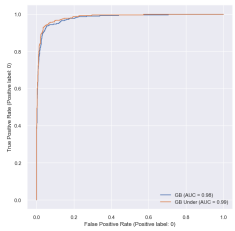
Aquí nuevamente se puede ver una gran mejoría en la clasificación. No solo hemos aumentado un 6% la accuracy total (llegando a 96%), sino que sobretodo hemos mejorado muchísimo el rendimiento a la hora de clasificar la clase Attrited Customers que tantas dificultades nos ha ido dando. Ahora, no solo tenemos un precision y recall totalmente equilibrados, sino que además tienen un valor de 89%, el cual ya se podría considerar excelente desde un punto de vista estadístico.



(a) GB Confusion Matrix



(b) GB Undersampled Conf. Mat.



(c) Curva ROC comparando GB con y sin Undersampling

Figure 11: Grafos de Gradient Boosting

Aquí vemos como el undersampling nos ha reducido a la mitad la cantidad de falsos negativos (aumentando el recall más o menos un 10%). Por lo tanto, ha mejorado de una forma bastante significativa el rendimiento del modelo. Observando la curva ROC de los dos modelos, vemos también como el GB con Undersampling (en naranja) presenta una curva ligeramente mejor y con más AUC. Por lo tanto, este modelo es objetivamente preferido a la alternativa sin undersampling. Además de ser el más prometedor por el momento.

6.4 Random Forest

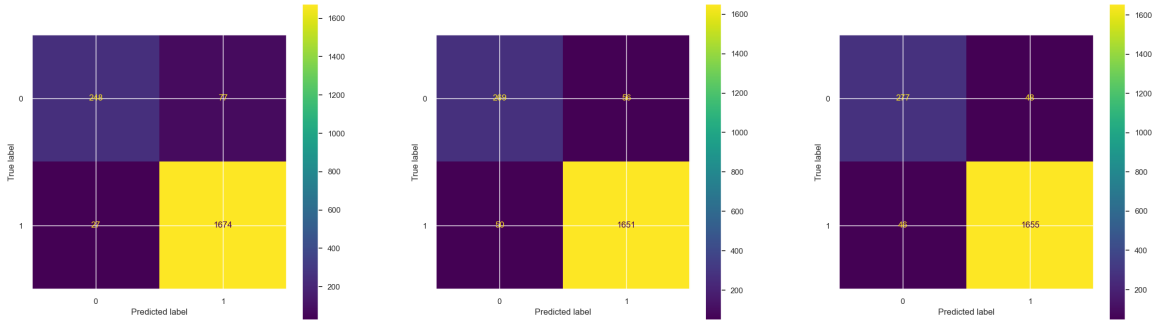
Random Forest es una técnica también bastante común para tratar problemas con relaciones no lineales con Decision Trees. Se basa en la capacidad de decisión intuitiva que tienen los DTs para combinar la respuesta de varios de éstos para, con suerte, dar un resultado un poco más elaborado gracias a este sistema de votación/aporte grupal. En este problema particular es especialmente interesante pues los Decision Trees, tal y como hemos comentado, permiten observar el problema desde un punto de vista muy intuitivo, cosa que nos permite analizar las decisiones tomadas y ver qué áreas afectan más a la salida de clientes.

Para el caso de Random Forest hemos hecho varios tests con y sin undersampling y con la variante de Weighted Random Forest (para sustituir el undersampling en la labor de equilibrar las clases). Los mejores resultados obtenidos son con Weighted Random Forest:

Reporte de Clasificación

	precision	recall	f1-score	support
Attrited Customer	0.86	0.85	0.85	325
Existing Customer	0.97	0.97	0.97	1701
accuracy			0.95	2026
macro avg	0.91	0.91	0.91	2026
weighted avg	0.95	0.95	0.95	2026

Vemos que aunque tiene un rendimiento similar al Gradient Boosting, no termina de llegar al mismo nivel de precision y recall sobretodo en lo que la clase Attrited Customers se refiere. Por lo tanto, Gradient Boosting sigue siendo el mejor modelo de los dos, con Weighted Random Forest como cercana segunda posición (lo cual nos da ya dos candidatos idóneos para hacer un ensamble: Undersampled Gradient Boosting y Weighted Random Forest, como veremos en próximos apartados).



(a) RF Confusion Matrix (b) RF Under. Conf. Mat. (c) Weighted RF Conf. MAt.

Figure 12: Comparativa de Random Forests: Con y sin undersampling y weighted

Observando estas gráficas vemos como la confusion matrix del Random Forest Undersampled no gana tampoco mucho beneficio sobre la de Random Forest sin undersampling, simplemente cambiando el ratio de precision y recall (Random Fores Undersampled tiene menos falsos positivos y por contra muestra más falsos negativos en una proporción bastante equivalente). Por otro lado, Weighted Random Forest sí que presenta una mejoría reduciendo tanto los falsos positivos (mejor precisión) como los falsos negativos (mejor recall). Esto es lógico pues Weighted Random Forests permite al modelo equilibrar los pesos según vaya aprendiendo cuál es la mejor configuración, cosa que ajusta mejor el equilibrio ótpimo de representación entre las 2 clases.

7 Comparativa Modelos

Después de desarrollar diversos modelos por separado comparamos la eficiencia de los mismos para ver que hemos ido obteniendo paulatinamente ligeras mejoras con cada decisión que nos han llevado a obtener máximos resultados en todos los apartados.

Nuestra tabla de resultados incluye muchas métricas diferentes para poder comparar todos los aspectos de los modelos. Al final por como han ido saliendo las cosas comparando únicamente

	test acc	precision score (W)	recall score (W)	f1 score (W)	ROC AUC	True Positives	False Positives	False Negatives	True Negatives
Voting GB1+RF CW	0.966	0.966	0.966	0.966	0.986	1666.0	34.0	35.0	291.0
Stacking GB1+RF CW	0.965	0.965	0.965	0.965	0.986	1669.0	39.0	32.0	286.0
Gradient Boosting_Undersampled	0.964	0.964	0.964	0.964	0.985	1664.0	36.0	37.0	289.0
Gradient Boosting	0.954	0.953	0.954	0.953	0.981	1670.0	63.0	31.0	262.0
Random Forest CW	0.954	0.953	0.954	0.954	0.983	1655.0	48.0	46.0	277.0
Random Forest	0.949	0.947	0.949	0.947	0.983	1674.0	77.0	27.0	248.0
Random Forest Undersampled	0.948	0.947	0.948	0.947	0.981	1651.0	56.0	50.0	269.0
DTree under samp	0.940	0.941	0.940	0.940	0.960	1638.0	58.0	63.0	267.0
DTree base	0.940	0.940	0.940	0.940	0.938	1638.0	59.0	63.0	266.0
DTree over samp	0.937	0.940	0.937	0.938	0.948	1622.0	49.0	79.0	276.0
RBF SVM binary	0.903	0.905	0.903	0.904	NaN	1592.0	88.0	109.0	237.0
Logistic	0.896	0.888	0.896	0.885	0.906	1658.0	168.0	43.0	157.0
polynomial SVM binary	0.890	0.910	0.890	0.896	NaN	1532.0	54.0	169.0	271.0
GNB	0.868	0.868	0.868	0.868	0.849	1568.0	134.0	133.0	191.0
Logistic Balanced	0.817	0.880	0.817	0.835	0.909	1387.0	57.0	314.0	268.0
linear SVM binary	0.810	0.882	0.810	0.831	NaN	1368.0	51.0	333.0	274.0

Figure 13: Todos los datos principales recogidos de las alternativas desarrolladas

Podemos apreciar que finalmente incluidos ensambles que comentaremos en el modelo final paso a paso vamos obteniendo mejores resultados. Se puede apreciar que con modelos lineales no somos del todo capaces de superar la marca del porcentaje de acierto de 90% comodamente debido a las limitaciones que presentan los modelos en un conjunto de datos de tan alta dimensionalidad dónde la no linealidad juega un papel importante. Esos 80% largos no parecen tan graves pero si observamos a la cantidad de ciertos negativos que obtenemos deja en evidencia que se nos escapa la mayoría de aquellas personas que se van a ir. Esto lo podemos compensar haciendo que los modelos como Logistic Regression balanceen las clases, obteniendo mejor recolección de verdaderos negativos. Si bien en detrimento de la precisión general pues se nos cuela un número significativo de falsos negativos.

Si damos un salto a los modelos no lineales comenzamos a ver un poco lo mejor de ambos mundos, Obteniendo el deseado salto pasado del 90% de acierto general y produciendo muy respetables (y pequeños) casos de falsos negativos y falsos positivos sin sacrificar tanto la cantidad de verdaderos negativos que somos capaces de capturar. Todo esto debido a la nueva habilidad de apreciar aquellas relaciones no lineales más costosas. Explorar con under sapmling de la clase principal y over sapmling de la clase minoritaria nos ha hecho ver que el undersampling de la principal es la mejor opción. Esto asienta la base de los próximos modelos basados en arboles. A continuación pasamos a Random Forest y Gradient Boosting. Los cuales mejoran las limitaciones de del anterior modelo. Uno aumentando la viabilidad construyendo múltiples arboles independientes aleatorios que aumentan la

robustez general (RF). Y otro obteniendo un enfoque secuencial que se va corrigiendo y se focaliza en aquellos errores (GB). Ambos obtienen resultados excelentes que cada uno tiene sus fortalezas individuales. Gradient Boosting obtiene un mejor rendimiento en general con su proceso iterativo. Pero la ejecución undersampleada de Gradient Boosting (balanceando mejor las clases) nos permite explorar de forma exhaustiva el dataset para obtener el máximo de verdaderos negativos del proyecto entero el cual como hemos ido diciendo es nuestro máximo objetivo empresarial.

Si adicionalmente los combinamos obtenemos lo mejor de ambos mundos con esa eficiencia generalizada de Random Forest balanceado y esa recolección de verdaderos negativos del Gradient Boosting con undersampling. Esto es interesante pues se complementan correctamente para mezclar sus mejores resultados y nos permiten alcanzar máximos en todas las formas de evaluación diferentes.

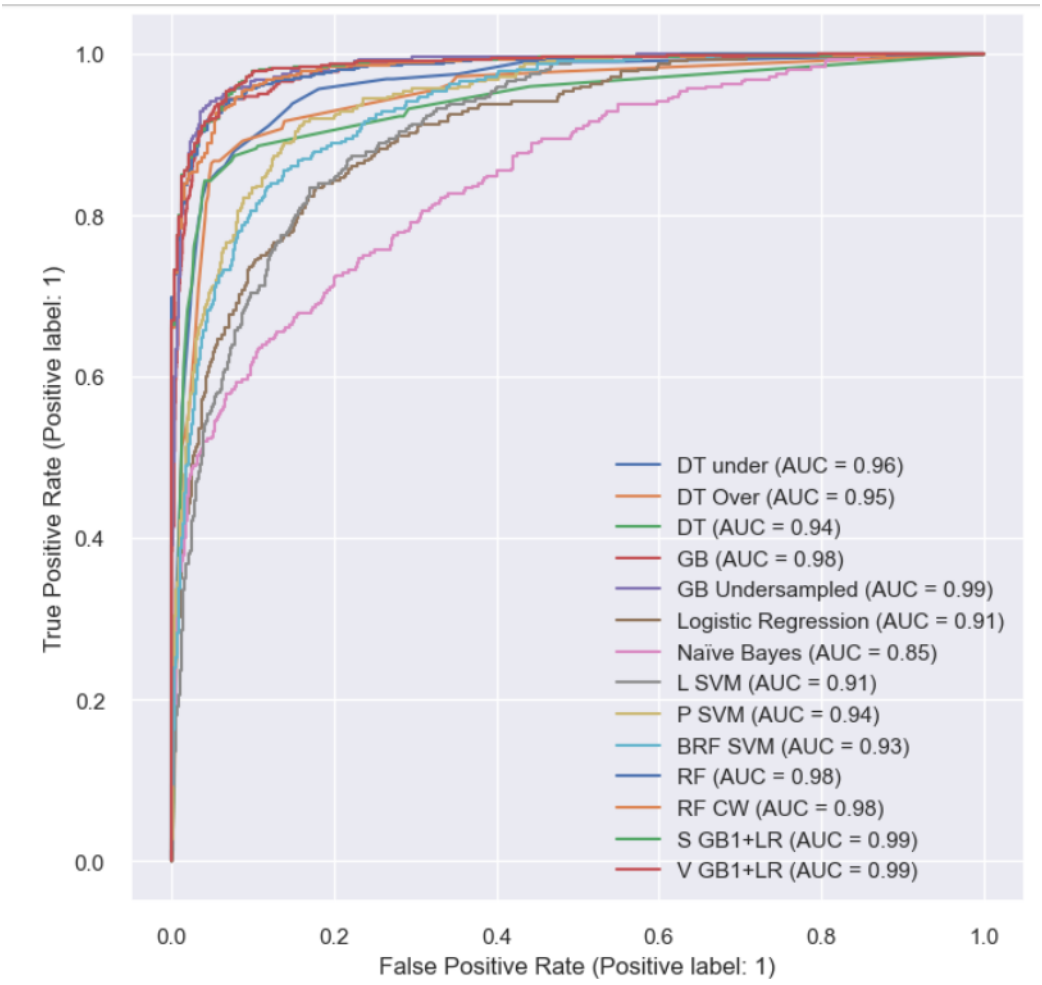


Figure 14: Curvas ROC de todos los modelos

En la curva ROC comparando todos los modelos sobre el valor negativo, pues este ha sido desde el principio el que más ha mejorado o necesitado mejora, podemos apreciar muchas cosas. En primer lugar, como los modelos no lineales suponen un claro salto de eficiencia y modelos tan sencillos como Naïve Bayes en este caso no suponen competencia real. Cuanto más nos acercamos a los modelos no lineales complejos más empinada es la curva y por lo tanto la tasa de falsos positivos respecto a los negativos es más baja. Esto también se puede ver en los datos de ROC AUC de arriba. Esto tiene sentido por ser esta la clase que para identificarla requería de explorar bien las relaciones de distancia no lineal que habia en los datos.

8 Modelo final

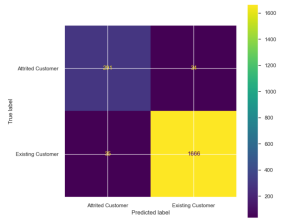


Figure 15: Matriz de confusión de voting classifier

Por lo tanto como hemos ido comentando nuestro modelo final propuesto es un ensamble entre Gradient Boosting con unersampling y Random Forest balanceado el cual explora las fortalezas de ambos las cuales se complementan para producir unos resultados muy competitivos. Resultado del proceso iterativo de pruebas que ha ido sucediendo durante el proyecto.

Este modelo aprovecha tanto el proceso iterativo de selección de gradient boosting con el preproceso adicional como la aleatoriedad y balanceado de Random Forest. Utilizamos tanto VotingClassifier como StackingClassifier y ambos estan bastante reñidos y obtienen más o menos la misma mejora. El primero combina mediante diferentes tipos de votación las predicciones de cada modelo haciendo que operen de forma independiente y se junten mediante el voto.

El segundo combina las predicciones mediante el entrenamiento de un meta-modelo entrenado con las predicciones de ambos. Estos dos enfoques són muy diferentes pero obtienen un resultado similar, siendo voting el que captura más clientes fugitivos con unos muy respetables 291 (90% de precisión). Por este motivo y como los dos están muy reñidos en todos los otros aspectos lo elegimos finalmente.

Figure 16: Matriz de conf. stacking classifier

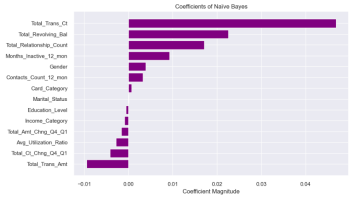
9 Analisis de interpretabilidad

Ahora analizaremos de forma conjunta que atributos son más importantes para los modelos y que diferencias hay entre los mismos.

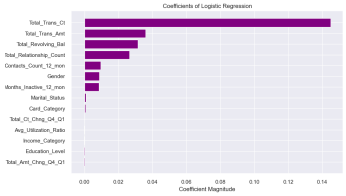
De esta manera veremos que variables son más o menos informativas, cuales aportan a los modelos y cuales reducen su capacidad de predicción. Esto se hace permutando las variables una a una y viendo el impacto que tiene esto en la eficiencia del modelo.

9.1 Atributos importantes

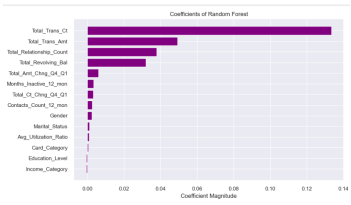
Nuestro objetivo principal era ver si ese salto de eficiencia encontrado en los modelos no lineales radicaba de una mejor elección de importancias de los atributos que utiliza (reduciendo la presencia de aquellos que tal vez produzcan ruido o escaso interés).



(a) Gaussian Naïve Bayes Importance



(b) Logistic Regression Importance (expone el caso de importancia lineal)



(c) Random Forest Importance (expone el caso de importancia de arbol)

Figure 17: Importancias de exponentes

Un patrón general que todos comparten (Excepto Naïve Bayes) es estar encabezados por *Total_Trans_Ct*, pues este esta altamente relacionado con la variable objetivo.

Después *Total_Trans_Amt* esta la segunda en casi todos cosa que es altamente lógica por estar esta muy correlacionada con *Total_Trans_Ct*. Esto en nuestro caso tiene sentido pues realmente el número discreto de transacciones es un mejor indicador que la cantidad de dinero total que hay en estas transacciones, pues una persona con un perfil económico bajo hace transacciones pequeñas pero es activo. Después vans seguidos de *Total_Relationship_Count* y *Total_Revolving_Bal* de muy cerca en posiciones intercambiables.

Después de este punto común la diferencia entre métodos lineales y no lineales es como de empinado es el descenso en los coeficientes. Los lineales suelen estar afectados en mayor medida sobre los minoriatrios y los no lineales solo se ven muy perturbados por el primero y despues hay un gran salto a los siguentes que en el cuarto o quinto valor es despreciable ya.

Esto indica que muchas de las features no acaban de aportar lo suficiente para ellos pues permutarlas no tiene impacto. En futuros proyectos se podría mirar de seleccionar otras variables.

9.2 Naïve Bayes como ejemplo anómalo

En ese sentido, como hemos ido pudiendo observar en la visualización de las gráficas en la explicación de los modelos anteriores podemos ver que los modelos probabílisticos lineales cómo es Naïve Bayes tiene una de las tablas de importancia más diferentes de las que encontraremos después. Estos se debe a que este modelo no interpreta muy bien las relaciones entre las variables y asigna probabilidades acorde de ciertas distribuciones. De esta manera encontramos que las importancias encontradas son atípicas en orden. Principalmente Total_Trans_Amt el cual se presenta como coeficiente negativo lo que quiere decir que las permutaciones del mismo ofrecen mejores modelos cuando en siguentes casos suele ser al revés.

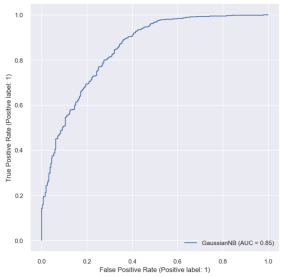


Figure 18: Roc gnb

Esto como hemos dicho se debe a que el método de clasificación de Naive Bayes se basa en el teorema de Bayes y asume independencia condicional entre cada par de características. Además de que nuestro conjunto presenta fronteras de decisión no lineales, cosa que puede afectar la capacidad del clasificador Naive Bayes para modelar correctamente dichas fronteras.

Produciendo esos valores anómalos que nos dejan un ROC Curve no tan bueno como podría ser.

9.3 Gradient Boosting como ejemplo relevante

En el Gradient Boosting es un ejemplo interesante de lo que serían los atributos importantes del tipo no lineal. Esto es porque se identifica perfectamente con el orden que suelen seguir y su curva desciende rápidamente indicando que aprende principalmente de los primeros y que el resto no le afectan enormemente.

Podemos apreciar también cómo cambia tras hacer Undersampling no cambia en absoluto las gráficas pues realmente el modelo aunque trabaje con un dataset alterado es el mismo y obtiene más o menos las mismas prioridades. Solamente que con la mejor representación de clases el sesgo de la clase mayoritaria tiene menos efecto y funciona mejor. El under sampling da algo más de importancia a algunas de las últimas variables.

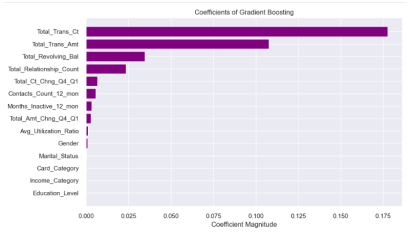


Figure 19: GB sin Undersample

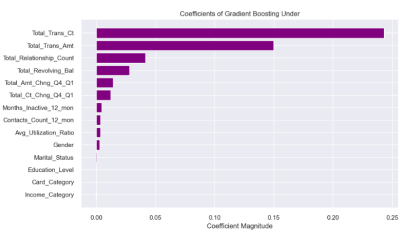


Figure 20: GB con Undersample

10 Conclusiones

10.1 Autoevaluación

Consideramos que hemos hecho una buena gestión inicial del trabajo y de la labor de repartir las tareas. Hemos aportado de una forma equilibrada tanto opinión sobre qué modelos usar como a la hora de implementar y luego comentar los resultados obtenidos. Si hay algo que nos podemos criticar a nosotros mismos es que hemos pecado un poco de "feature creep" y al final hemos ido probando diferentes variaciones de modelos y se nos ha ido un poco la extensión del trabajo de las manos. Esto ha terminado dificultando un poco la recta final y que no llegaramos con tanto margen de tiempo y espacio como nos hubiera gustado. Aún y así, consideramos que hemos aprendido mucho de esta práctica y estamos satisfechos con el proceso desarrollado.

10.2 Conclusiones Científicas

Una de las cosas más inmediatamente aparentes de nuestro dataset era el hecho que había muchas variables con poca correlación con la variable objetivo y tampoco con las otras variables del dataset. Por lo tanto, eran irrelevantes y de hecho en algunos casos dificultaban el análisis (por ejemplo empeorando la visualización de técnicas como T-SNE). De hecho, las principales variables que influían en la predicción eran siempre las mismas: Total_Trans_Ct, Total_Trans_Amount, Total_Relationship_Count... Es decir, variables que tienen relación con los movimientos bancarios y la cantidad de dinero movido. Por lo tanto, sugeriríamos al banco en cuestión recolectar menos datos (para ahorrar espacio) o recolectar de diferentes y ver como influyen en las predicciones.

Finalmente, estamos muy satisfechos con el modelo final (el ensamble de Gradient Boosting Undersampled + Weighted Random Forests). Puesto que hemos obtenido el excelente resultado de 96.6% de accuracy global en ese modelo, con una f1-score de alrededor de 90% para la clase Attrited Customers, que para ser una clase que parecía estar tan intermezclada con la otra, es algo de lo que estar satisfecho. Además, gracias a la comparativa de los otros modelos que se han ido experimentando, podemos afirmar la relativa optimalidad de este modelo final y sin señales de overfitting, pues durante el proceso se ha ido analizando los test scores de las CV para asegurarlo. Por lo tanto tenemos un modelo bien ajustado, con buenos rendimientos y sin señales de sufrir overfitting.

10.3 Conclusiones Personales

Este trabajo nos ha servido mucho para terminar de consolidar el temario de la asignatura. Hemos aprendido a como escoger, optimizar y comparar distintos modelos. Además creemos que el dataset era especialmente interesante desde un punto de vista de aprendizaje. Hemos tenido que lidiar con eliminación de features innecesarias, missing values (mediante imputación knn) y estandarización de variables. Además, el dataset estaba muy desbalanceado, cosa que nos ha hecho tener que considerar técnicas de mitigación de este suceso. En general, estamos muy contentos con esta práctica, consideramos que no aplicamos técnicas de forma aleatoria sino que vamos enlazando los descubrimientos que vamos haciendo de una forma elegante y conexa, iterativamente aplicando nuestros nuevos descubrimientos para avanzar el trabajo.

10.4 Posibles extensiones y limitaciones

El proyecto se podría extender haciendo uso de MLPs para tanto visualización como clasificación de los datos. Puesto que es otra técnica no lineal y bastante versátil, podría mostrar resultados bastante buenos. En cuanto a limitaciones, debido a la gran cantidad de modelos que hemos explorado y el nivel de detalle que hemos optado, quizás nos ha fallado un poco más el hecho de pulir el notebook con el código fuente con comentarios y información sobre el procedimiento. Pues hemos optado por comentar los resultados de forma detallada en la documentación ya que el procedimiento experimental ha sido relativamente mecánico y repetitivo.

11 Referencias

Bibliografia General

[1] Bank Customer Churn Prediction using Machine Learning by Yosafat jul 7 2023, via medium.com https://medium.com/@crypter70/bank-customer-churn-prediction-using-machine-learning-514516ecf82e

[2] Zhenkun Liu, Ping Jiang, Koen W. De Bock, Jianzhou Wang, Lifang Zhang, Xinsong Niu, *Extreme gradient boosting trees with efficient Bayesian optimization for profit-driven customer churn prediction, Technological Forecasting and Social Change,* Volume 198, 2024, 122945, ISSN 0040-1625, https://www.sciencedirect.com/science/article/pii/S0040162523006303

[3] Yaya Xie, Xiu Li, E.W.T. Ngai, Weiyun Ying, *Customer churn prediction using improved balanced random forests, Expert Systems with Applications,* Volume 36, Issue 3, Part 1, 2009, Pages 5445-5449, ISSN 0957-4174, https://www.sciencedirect.com/science/article/pii/S0957417408004326

[4] Commun.Fac.Sci.Univ.Ank.Ser. A1 M ath. Stat. Volume 70, Number 2, Pages 827-836 (2021) *PREDICTING CREDIT CARD CUSTOMER CHURN USING SUPPORT VECTOR MACHINE BASED ON BAYESIAN OPTIMIZATION*, ISSN 1303-5991 E-ISSN 2618-6470 https://dergipark.org.tr/en/download/article-file/1646899

[5] Andres H.G., Ankit Gupta, Thomas Konstantin, *Credit Card customers, Predict Churning customers,* Pages 827-836 (2021) *PREDICTING CREDIT CARD CUSTOMER CHURN USING SUPPORT VECTOR MACHINE BASED ON BAYESIAN OPTIMIZATION*, ISSN 1303-5991 E-ISSN 2618-6470 https://dergipark.org.tr/en/download/article-file/1646899