CIMAT

# Técnica de clasificadores: *Bank Marketing Dataset*

## Introducción a Ciencia de Datos.

**Autores:** Brain de Jesús Salazar García [†], Rodrigo Gonzaga Sierra[†],Cesar Moises Avila Montes[†]
**Profesor:** Dr. Marco Antonio Aquino López [†]

Centro de Investigación en Matemáticas A. C.[†]

**RESUMEN:**
En este reporte se incluye la comparación de múltiples técnicas de clasificación—Naive Bayes Gaussiano, Análisis Discriminante Lineal (LDA), Análisis Discriminante Cuadrático (QDA), Criterio de Fisher y K-NN—aplicadas al conjunto de datos *Bank Marketing* para predecir la suscripción de clientes a depósitos a plazo fijo. Tras un preprocesamiento que incluyó codificación de variables categóricas y escalado de numéricas, se evaluó el desempeño de cada modelo considerando el desbalance inherente de los datos (85 % "no" vs. 15 % "sí").

# Índice

# 1. Introducción

En este proyecto compararemos diversas técnicas de clasificación (Naive Bayes, LDA, QDA, Fisher y $k$-NN), aplicándolas en datos provenientes de campañas de marketing de una institución bancaria. Presentaremos una descripción de los datos y su preprocesamiento. Después se discuten las particularidades de cada modelo y su aplicación. Finalmente, una vez realizados los modelos, se reportan los tests correspondientes para comparar los clasificadores.

# 2. Exploración inicial

En este proyecto trabajaremos con la base de datos "Bank Marketing [Dataset] (2014) " del repositorio de Machine Learning de la UCI, [Moro and Cortez, 2014]. Los datos provienen de una campaña de marketing directo (mediante llamadas telefónicas) realizada de mayo de 2008 a noviembre de 2010 por una institución bancaria portuguesa. Se espera ser capaz de predecir si un cliente se suscribirá a un depósito bancario a plazo fijo.

Para cada cliente contamos con 16 variables características que contienen su información personal, sus datos bancarios y el resultado de campañas de marketing previas; y una variable binaria objetivo que indica si el cliente se suscribió o no al depósito. Los detalles del tipo de variables y sus descripciones pueden consultarse en la tabla 1. Se cuenta con 41211 entradas de clientes.

El objetivo de este proyecto es comparar las distintas técnicas de clasificación mencionadas. Si nuestro objetivo fuese predecir la variable objetivo para clientes nuevos, deberíamos tener cuidado con variables características que pueden influir mucho en la variable objetivo con las que no contamos para un cliente nuevo, como lo es la duración de la última llamada con el cliente.

| **Nombre de la Variable** | **Rol** | **Tipo** | **Demográfico** | **Descripción** |
|---|---|---|---|---|
| edad | Característica | Entero | Edad | |
| trabajo | Característica | Categórica | Ocupación | tipo de trabajo: 'admin.', 'obrero', 'emprendedor', 'empleada doméstica', 'gerente', 'desempleado', 'autónomo', 'jubilado', 'servicios', 'estudiante', 'técnico', 'desconocido' |
| estado_civil | Característica | Categórica | Estado Civil | 'divorciado', 'casado', 'soltero', 'desconocido' (divorciado incluye viudo) |
| educación | Característica | Categórica | Nivel Educativo | 'basic.4y', 'basic.6y', 'basic.9y', 'bachillerato', 'analfabeto', 'curso.profesional', 'universitario', 'desconocido' |
| incumplimiento | Característica | Binaria | | ¿tiene crédito en incumplimiento? (sí/no) |
| saldo | Característica | Entero | | saldo promedio anual (€) |
| hipoteca | Característica | Binaria | | ¿tiene préstamo hipotecario? (sí/no) |
| préstamo | Característica | Binaria | | ¿tiene préstamo personal? (sí/no) |
| contacto | Característica | Categórica | | tipo de contacto: 'celular', 'telefono' |
| día_de_la_semana | Característica | Fecha | | último día de la semana de contacto |
| mes | Característica | Fecha | | último mes de contacto ('jan', 'feb', ..., 'dec') |
| duración | Característica | Entero | | duración del último contacto, en segundos |
| campaña | Característica | Entero | | número de contactos durante esta campaña para este cliente (incluye el último contacto) |
| pdias | Característica | Entero | | número de días que pasaron desde la última ves que se contacto al cliente para una campaña previa |
| anterior | Característica | Entero | | número de contactos realizados antes de esta campaña |
| presultado | Característica | Categórica | | resultado de campañas previas de marketing ('fracaso', 'noexistente', 'éxito') |
| y | Objetivo | Binaria | | ¿El cliente se suscribió al depósito a plazo fijo? (sí/no) |

Cuadro 1: Tabla de variables

# 3. Preprocesamiento

De acuerdo con [James, 2013], codificar y escalar son transformaciones fundamentales que convierten datos "en crudo" en representaciones matemáticamente sólidas que permiten a los algoritmos de ML funcionar correctamente y ser eficientes. Por lo que se realizó la siguiente **Codificación de Variables Categóricas:**

- **One-Hot Encoding:**

  Para variables nominales como `job, marital, contact, month, day_of_week, poutcome.` Debido a que estas variables no tienen orden inherente, por lo que One-Hot Encoding evita introducir relaciones ordinales artificiales. Además, que La mayoría de algoritmos de ML trabajan mejor con representaciones numéricas, y One-Hot preserva la naturaleza categórica sin imponer orden (mantiene equidistancia entre categorías).

- **Label Encoding:**

  Para variables binarias/ordinales como `default, housing, loan, education, y`. Estas suelen tener un orden natural como educación:

$$\text{Educación} \rightarrow \begin{cases} -1 & \text{unknown} \\ 0 & \text{illiterate} \\ 1 & \text{basic.4y} \\ 2 & \text{basic.6y} \\ \vdots & \vdots \\ 6 & \text{university.degree} \end{cases}$$

  Label Encoding es resulta ser eficiente por reducir la dimensionalidad y es computacionalmente más eficiente que One-Hot Encoding.

Ahora sobre el **escalamiento de Variables Numéricas,** se realizó lo siguiente:

- **z-score**: Este escalamiento transforma los datos para tener media 0 y desviación estándar 1:

$$z = \frac{x - \mu}{\sigma}$$

  donde $\mu$ es la media y $\sigma$ la desviación estándar. Esto ayuda a algoritmos sensibles a la escala. Observe que las variables como `age` (20-90) y `euribor3m` (0-5) están en escalas muy diferentes, lo que podría sesgar modelos como SVM o KNN.

Por último se implementó un **Manejo de Valores Especiales**, debido a la variable `pdays = 999`, pues el valor 999 indica "no contacto previo", que es información cualitativamente diferente de valores numéricos y los categóricos:

$$\text{pdays} = \begin{cases} n & \text{Días desde último contacto} \\ 999 & \text{Sin contacto previo (valor especial)} \end{cases}$$

Para esto resulta factible definir una variable binaria: `previous_contact` que capture mejor esta información semántica:

$$\text{previous\_contact} = \mathbb{I}(\text{pdays} \neq 999) = \begin{cases} 1 & \text{Con contacto previo} \\ 0 & \text{Sin contacto previo} \end{cases}$$

Se tiene la siguiente tabla obtenido del comando head().

| default_encoded | housing_encoded | ... | poutcome_nonexistent | poutcome_success |
|---|---|---|---|---|
| 0 | 0 | ... | True | False |
| 1 | 1 | ... | True | False |
| 0 | 2 | ... | True | False |
| 0 | 0 | ... | True | False |
| 0 | 0 | ... | True | False |

Cuadro 2: Tabla de datos preprocesados

Los datos están listo para ser modelados.

# 4. Modelación

A continuación, se realiza la implementación de algunos clasificadores y se discute su desempeño analizando distintas características como, exactitud, sensibilidad, especificidad, (F1, AUC) mediante técnicas de validación apropiadas. Dichos clasificadores, tienen como objetivo modelar y predecir los valores de la variable $Y = \{1, 0\}$ la cual representa la decisión del cliente de contratar un deposito de plazo bancario($Y = 1$ para si y $Y = 0$ para no), a partir de los valores de las covariables dadas en las tablas.

## 4.1. Naive Bayes Gaussiano

El primer modelo que se ajusta a estos datos es el Naive Bayes con distribución gaussiana. Recuerde que en este modelo, una supuesto fuerte es el de independencia condicional de las covariables dada la clase. En este caso, se sabe que covariables como la tasa de interés, tasa de empleo y el indice de confianza del consumidor están tienen una estrecha relación en el mundo real. Además, se pueden considerar indicadores relevantes sobre el ambiente económico general, lo que también puede influir en la decisión del cliente, por lo que no resulta prudente eliminar esta información del data set. Un inconveniente más general podría ser el des balance de la muestra, ya que de todo el data set, el 85 % es de Y="no", y solo el 15 % corresponde a casos de Y="si". Esto podría llevar a estimaciones con mayor varianza para el subconjunto pequeño, así como a problemas para aprender el comportamiento de este caso. A continuación, la figura (1) muestra la matriz de confusión, en ella se puede ver que la capacidad para predecir valores de Y="no", es significativamente mayor que para el otro caso, esto puede ser un síntoma del desbalanceo de la muestra, además para Y="si"tuvo problemas al momento de capturar su comportamiento ya que la mayoría de los casos reales de Y="si", el modelo se equivoco al predecir su valor.

Se puede ver que el porcentaje total de predicciones correctas es de 0.8698713279922311 la cual parece ser alto, esto se debe a que el modelo capturo bien el comportamiento de la población mayoritaria, sin embargo con la población pequeña no tuvo el mismo comportamiento

Por otro lado la precisión,dada por la razón de ciertos verdaderos sobre ciertos verdaderos más ciertos falsos es 0.876460038757151 lo cual se puede explicar ya que es el promedio pondera de la exactitud en cada clase, sin embargo, este valor alto puede ser explicado ya que para la clase 0 el modelo tiene una exactitud de 93.3 % mientras que ara la clase 1 tiene una exactitud de 43.0 %.

La sensibilidad, que explica el porcentaje acertado de valores reales, es 0.8698713279922311, nuevamente este valor está sesgado por la cantidad de aciertos que se tiene en la clase 0.

Por último, observe que el F1-score está dado por 0.8729649875102999, esta es una medida del balance que tiene el modelo, además está dada por un promedio ponderado de la medida de cada clase, por lo que este valor está influenciado principalmente por la clase 0.

Observe que a la hora de entrenar el modelo, unos de los parámetros que se deben estimar son las aprioris, estos valores son estimados por el porcentaje relativo de cada población con respecto a la población total. Observe además que si estos valores están muy desbalanceados, el clasificador favorecerá fuertemente a la clase mayoritaria. Por ello, una propuesta para tratar de mitigar el

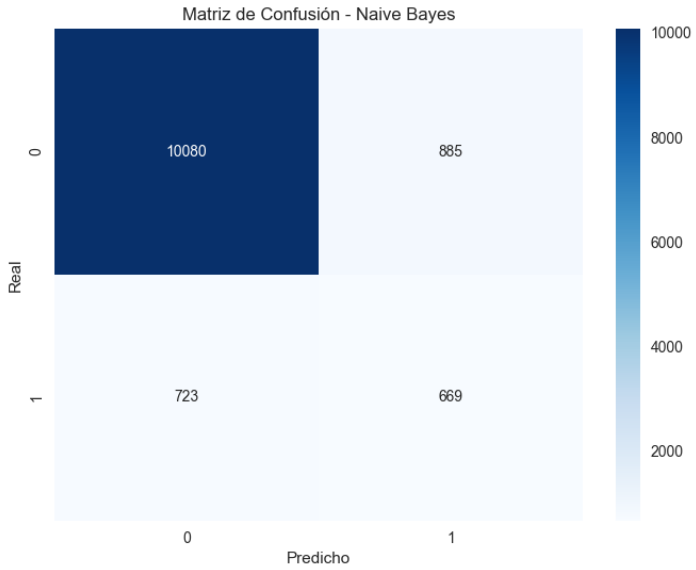



Figura 1: Matriz de confusión del modelo Naive Bayes.

efecto del desbalance de la muestra es ingresar aproris igualitarias, 0.5, para cada muestra. Si se entrena el modelo con está variante se tiene la siguiente matriz de confusión.

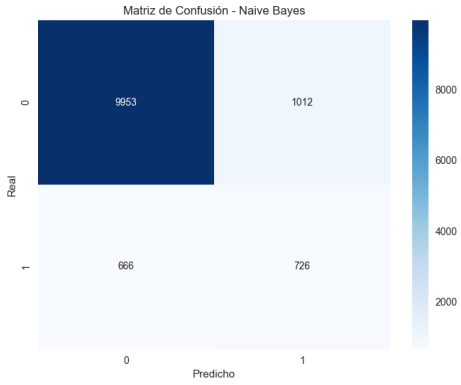
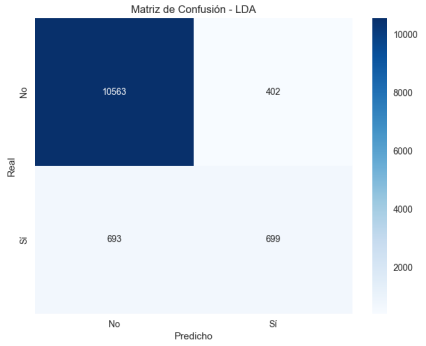



Figura 2: Matriz de confusión con aprioris igualitarias.

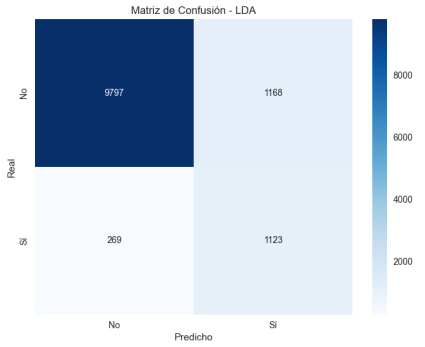
Observe en la figura (2) que con está variante el número de aciertos en la clase 1 aumentó, sin embargo el número de aciertos en la clase 0 disminuyó. Además los valores de las métricas de evaluación no cambiaron mucho, teniendo valores de F1 de 0.8706236293009308 y de sensibilidad de 0.8642065226187586.

## 4.2. Linear Discriminant Analysis

Otra propuesta para modelar este data set es Linear Discriminant Analysis. En este caso, ya que se asume una distribución normal multivariada como distribución conjunta para cada clase con matriz de varianzas y covarianzas general, es decir que esta no se estima solo de una clase si no de todo el conjunto de datos, se espera que se pueda capturar mejor la interacción entra variables con estrecha relación como la tasa de interés, tasa de empleo y el indice de confianza del consumidor. El desbalance de la muestra se atenúa un poco, ya que el efecto del desbalance se ve unicamente en el termino log $\frac{\pi_1}{\pi_2}$, el cual para este caso resulta ser pequeño. Sin embargo, el umbral de la decisión dependerá mayormente del termino $\frac{1}{2}\left(\mu_2+\mu_1\right)^\top \Sigma^{-1}\left(\mu_2-\mu_1\right)$, el cual estará estimado por los datos. A continuación, la figura (3) muestra la matriz de confusión, en ella se puede ver que la capacidad para predecir valores de Y="no", es significativamente mayor que para el otro caso, así como que para Y="si", tuvo problemas al momento de capturar su comportamiento. Sin embargo, el modelo LDA parece tener un mejor resultado que el modelo Naive Bayes, ya que aumentó el número de aciertos en la clase 1, y si bien disminuyó el número de aciertos en la clase 0, está dismiución es menor.




(a) Matriz de confusión del modelo LDA.




(b) Matriz de confusión para el modelo LDA con aprioris iguales.

Figura 3: Comparación de matrices de confusión para LDA con diferentes configuraciones de aprioris.

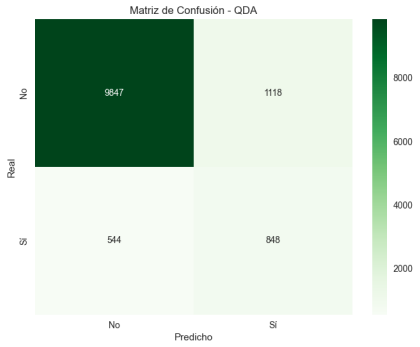
Se puede ver que el porcentaje total de predicciones correctas es de 0.91138625. Este valor es mayor que en el caso de Naive Bayes ya que el aumento de aciertos en la clase 1 es relativamente mayor que la perdida de aciertos en la clase cero. Por otro lado la precisión,dada por la razón de ciertos verdaderos sobre ciertos verdaderos más ciertos falsos es igual a 0.91138625880, lo cual se puede explicar por el aumento de aciertos en la clase 1. La sensibilidad, que explica el porcentaje acertado de reales positivos, está dado por 0.9113862588006798, y el F1-score está dado por 0.9067946884. Observe que en general, el porcentaje de aciertos de la clase 1 aumento significativamente, esto puede estar relacionado con el planteamiento del modelo, ya que en este modelo se está considerando la interacción entre cierta variables que tienen una estrecha relación por medio de su matriz de varianzas y covarianzas. Sin embargo, observe que la sensibilidad en la clase 1 sigue siendo bajo.

Observe que para el caso de aprioris iguales los resultados mejoran considerablemente. En la figura 3 se muestra la matriz de confusión para este caso.

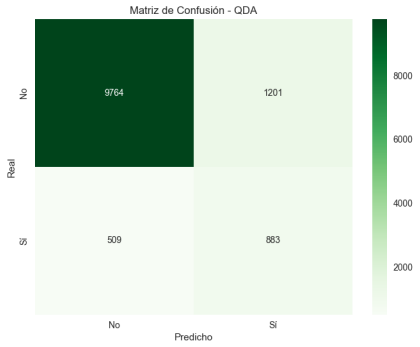
Con esta adaptación se tiene un mejor desempeño del modelo, ya que se detectan muchos más verdaderos positivos y mejora la sensibilidad de la clase 1. Sin embargo, aunque el modelos es más sensible para la clase 1, también se vuelve menos preciso.

## 4.3. Discriminante Cuadratica(QDA)

El modelo de dicriminante cuadratico tiene un planteamiento muy similar al modelo de discriminante lineal, con la variante de que la matriz de varianzas y covarianzas para las distribuciones serán distintas para cada clase. Esta modificación puede resultar en información más especifica para cada clase sin discriminar la relación entre covariables estrechamente relacionadas, la tasa de interés, tasa de empleo y el indice de confianza del consumidor. Sin embargo, al tener relativamente, pocos valores de la clase Y="si", las estimaciones pueden resultar no tan exactas y aumentar la varianza en las estimaciones. Si bien, es más flexible en su umbral discriminante, el comportamiento capturado para la muestra pequeña puede ser ruidosos. A continuación, la figura (4) muestra la matriz de confusión, en ella se puede ver que la capacidad para predecir valores de Y="no", es significativamente mayor que para el otro caso, así como que para Y="si"tuvo problemas al momento de capturar su comportamiento.




(a) Matriz de confusión del modelo QDA.




(b) Matriz de confusión del modelo QDA con aprioris iguales.

Figura 4: Comparación de matrices de confusión para QDA con diferentes configuraciones de aprioris.

Se puede ver que el porcentaje total de predicciones correctas es de 0.8894848, la precisión,dada por la razón de ciertos verdaderos sobre ciertos verdaderos más ciertos falsos es igual a 0.865501, la sensibilidad, que explica el porcentaje acertado de reales positivos, está dado por 0.865501 y por último, observe que el F1-score está dado por 0.8751890, esta es una medida del balance que tiene el modelo. Estos valores son razonables con el comportamiento de los datos, además observe que, introducir un matriz de vrianzas y covarianzas por clase ayudó a capturar mejor el comportamiendo de los datos de la clase 1, aumentando con esto el número de datos bien identificados de esta clase. Pareciera que esto perjudicó a la identificabilidad de los datos de la clase 0, sin embargo, esta disminución en realidad no es significativa en comparación al modelo anterior.

## 4.4. Criterio de Fisher

La idea principal para clasificar mediante el criterio de Fisher es pasar las covariables de un espacio de dimensión $d$ a un espacio de dimensión uno. Esto se puede hacer mediante la elección de un vector adecuado $a^T$, el cual maximizará la razón de la varianza entre clases sobre la varianza dentro de las clases. Observe que con este modelo, el desbalanceo de la muestra no parece ser un problema grave, ya que la estimación de la matriz de varianzas es global, además de que no se le está asignando ninguna distribución especifica a los datos, se sigue considerando la interacción de variables fuertemente relacionadas. A continuación, la figura (5) muestra la matriz de confusión, en ella se puede ver que la capacidad para predecir valores de Y="no", es significativamente mayor que para el otro caso, así como que para Y="si", tuvo problemas al momento de capturar su comportamiento.

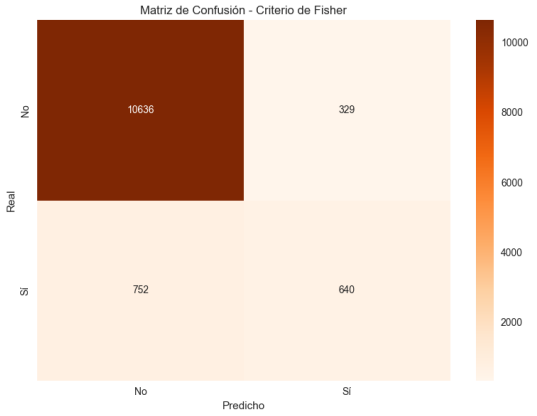



Figura 5: Matriz de confusión con el criterio de Fisher.

Este modelo no pareciera ser una buena opción para el caso en que se quiere encontrar a potenciales clientes que si adquieran la suscripción, ya que la mayoría de los que en realidad si adquirieron la suscripción, el modelo los predio como "no". Esto resulta más perjudicial, en un caso en el que se busque capturar la mayor cantidad de suscripciones posibles, es preferible hacer más intentos con la predicción de que si la compraran, a limitarse por malas predicciones. Si bien, el modelo de separación lineal de Fisher obtuvo un accuracy de 0.8616, equivalente a 86.16 % de clasificaciones correctas, la precisión alcanzó 0.8911 (89.11 %), indicando alta confiabilidad en las predicciones. la sensibilidad fue de 0.8616 (86.16 %), demostrando buena capacidad para detectar casos reales y el F1-score de 0.8731 (87.31 %) refleja un balance adecuado entre precisión y sensibilidad, hay que notar que estos valores están afectados por el desbalance de la muestra, ya que la clase cero es la responsable de la mayoria de los casos correctos.

## 4.5. K-NN

El modelo de k vecinos más cercanos consiste en proporcionar una etiqueta mediante las etiquetas de sus vecinos más cercanos. Una de las ventajas de este modelo es que no se asume alguna distribución en especifico para los datos además de que no se necesita estimar algún parámetro en especifico para la población pequeña. Sin embargo, la elección de un k adecuado se vuelve crucial, ya que el desbalance en la muestra puede hacer que para un k grande el modelo se incline demasiado por la población mayoritaria y no alcance a capturar el comportamiento de la población pequeña, mientras que para un k muy pequeño, el comportamiento estará muy influenciado por la población mayoritaria. Parece razonable asignar pesos por distancia ya que al tener una muestra desbalanceada, la población mayoritaria tendría cierta ventaja y se estaría despreciando el echo de que vecinos más cercanos tienen mayor probabilidad de ser de la misma clase. Además, observe no se puede utilizar cualquier métrica, ya que se tienen datos numéricos y categóricos, en este sentido Otra observación a este modelo es que se tiene un tamaño grande en la dimensión de los datos lo que influye a que las distancias no sean tan distintivas.

A continuación, la figura (6) muestra la matriz de confusión, en ella se puede ver que la capacidad para predecir valores de Y="no", es significativamente mayor que para el otro caso, así como que para Y="si", tuvo problemas al momento de capturar su comportamiento.

Se puede ver que el porcentaje total de predicciones correctas es de 0.888294, la sensibilidad, que explica el porcentaje acertado de reales positivos, está dado por 0.900461, por último, observe que el F1-score está dado por 0.8920978, esta es una medida del balance que tiene el modelo. En general, el resultado parece bastante balanceado, el número de falsos positivos disminuyo considerablemente, y parece que el modelo identifico bien a los elementos de la clase 1 ya que al tratar de identificarlos, acertó en la mayoría.

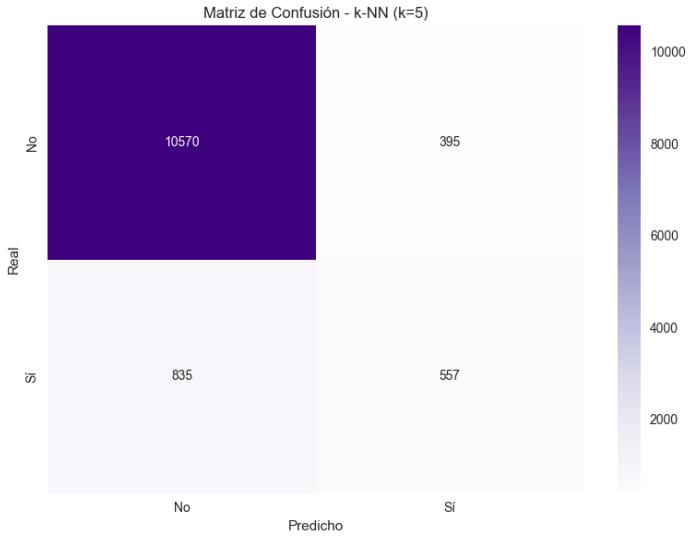



Figura 6: Matriz de confusión para el modelo K-NN.

# 5. Conclusiones

Una vez que se analizaron los resultados de los modelos, se pudo observar que hay casos en los que no se logra capturar de manera adecuada la dinámica de la población minoritaria, pero si se modela bien el comportamiento de la clase mayoritaria. En este sentido, una forma de definir que modelo resulta más conveniente usar, es definiendo de manera explicita la tarea que se quiere resolver. Por ejemplo, si se busca un modelo que identifique mejor a los clientes que si van a adquirir la suscripción del deposito a plazo, entonces una buena opción es el modelo LDA con aprioris iguales, ya que en el aumenta el número de aciertos en los casos de aceptación y aunque disminuye el número de de acierto para la clase 0, esta disminución no es significativa a comparación de el poder de clasificación que se gana respecto a la clase 1. Esto pensando en que interesa vender el mayor número de suscripciones, es decir, si bien al momento de tratar de predecir en la clase "si", se acierta aproximadamente la mitad, esto es preferible a descartar clientes que en realidad si iban a adquirir la suscripción.

Por otro lado, si lo que se busca es poder identificar mejor a los clientes que no van a adquirir la suscripción del deposito a plazo, entonces una opción viable es el modelo LDA el cual aprende mejor el comportamiento de la población que no adquiere la suscripción y tiene mayor precisión a la hora de identificarlos.

# Referencias

[Aquino López, 2024a] Aquino López, M. A. (2024a). Diapositivas del curso: Introducción a la ciencia de datos. Recuperado de: `https://github.com/maquinolopez/Ciencia_De_Datos/blob/main/Diapostivas/Presentation_7.pdf`. Accedido: 28 de septiembre del 2025.

[Aquino López, 2024b] Aquino López, M. A. (2024b). Diapositivas del curso: Introducción a la ciencia de datos. Recuperado de: `https://github.com/maquinolopez/Ciencia_De_Datos/blob/main/Diapostivas/Presentation_8.pdf`. Accedido: 28 de septiembre del 2025.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., Friedman, J., et al. (2009). The elements of statistical learning.

[James, 2013] James, G. (2013). An introduction to statistical learning with applications in r.

[Moro and Cortez, 2014] Moro, S., R. P. and Cortez, P. (2014). Bank Marketing. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K306.

**Contribuciones:**

Introducción y análisis exploratorio: Cesar.

Preprocesamiento y Naive Bayes Gaussiano: Rodrigo.

Linear Discriminant Analysis, QDA, Criterio de Fisher, K-NN: Brain de Jesús

Conclusiones: Rodrigo y Brain de Jesús.

Código en Python: Rodrigo y Brain de Jesús.