

Sistema inteligente para la segmentación de clientes con el fin de potencializar y optimizar la conversión de la campaña deposito financiero

Sebastián Herrera¹, Edison Neria¹, Sergio Rairán.¹

¹ Pontificia Universidad Javeriana, Bogotá CO 110231, Colombia

Resumen. Una institución bancaria portuguesa propone por medio de una campaña de marketing basada en llamadas telefónicas predecir si un usuario se suscribirá a un depósito a plazos. De cada usuario se tiene atributos como el trabajo actual, el nivel de educación, el incumplimiento financiero, etc. Actualmente, la institución realiza las llamadas sin clasificar ni priorizar clientes, como consecuencia se tiene un tiempo elevado en la labor, costos elevados e incomodidad en clientes. Se desarrollarla por medio de la técnica Redes Neuronales Artificiales (ANN por sus siglas en inglés Artificial Neural Network) un modelo de clasificación que buscará, basado en las características propias de cada cliente, predecir cual será la decisión de si adquiere el producto. La medición exitosa del modelo estará dada por un nivel de sensibilidad superior al 80%, y será validada por medio de la curva roc, buscando así aumentar el recaudo financiero de la compañía. La base inicial para el desarrollo contara con 16 variables que explicaran el tomar o no la campaña.

Palabras clave: Las Redes Neuronales Artificiales, ANN (Artificial Neural Networks), Smote – Tomek, Análisis de componentes principales, Random Forest.

1 Introducción

Una de las modalidades que tienen las entidades financieras para generar ingresos es la generación de títulos valores, entre los que se encuentran los Certificados de Depósito a Término (CDT). Estos productos, dependiendo del plazo y las condiciones ofrecidas por una entidad, pueden generar una renta para el inversor que puede ir desde el 2% hasta el 5.9% [1]. Estos CDT le permite al banco tener un mayor músculo financiero y mejorar su capacidad de colocación de préstamos en el mercado, sin embargo, el costo asociado a la comercialización de dichos productos se ve impactado por el mal manejo de los clientes actuales de las entidades. La falta de estrategias focalizadas a clúster de clientes específicos hace que el número de llamadas sea ineficiente generando un mayor número de llamadas, lo que estaría generando inconformidad y una mala percepción de los clientes respecto al banco.

Un banco de Portugal proporciona una base de clientes a los cuales se les aplicó una campaña de marketing basada en llamadas telefónicas con el fin que sus clientes se suscribieran a un depósito bancario a plazo fijo, donde la respuesta para la base sería “si” estar suscrito o “no” estar suscrito a la campaña[2].

En un mercado donde ser eficiente con el manejo de los recursos físicos, económicos y humanos es la prioridad, tener un sistema óptimo generará un nivel de costo bajo para la entidad, al igual que reducirá las molestias por colocación de cara a sus clientes, logrando con el mismo número de llamadas una mayor colocación de CDT, consiguiendo una mayor liquidez y subiendo los indicadores generales del banco.

2 Estado del arte

2.1 Certificados de depósito a termino

Un CDT es un certificado de depósito a término, donde se deposita cierta cantidad de dinero por un tiempo fijo en bancos comerciales, corporaciones de ahorro y vivienda, corporaciones financieras y compañías de financiamiento comercial. Estos serán mantenidos por 30 días o más, y este dinero no puede ser retirado antes que se haya cumplido el periodo pactado. Los CDT más comunes son a 30, 60, 90, 180 y 360 días. El interés ganado por cada depósito variará de acuerdo con la cantidad de dinero colocada y la duración del CDT.

El interés estará dado por la DTF, y para su cálculo se tendrá en cuenta lo siguiente:

- Recolectar para cada una de las entidades financieras descritas anteriormente (bancos, corporaciones financieras, corporaciones de ahorro y vivienda y compañías de financiamiento comercial), el valor de la tasa de interés que reconocen por los CDT a 90 días y la cantidad de recursos (dinero) que la gente tiene depositados en CDT a 90 días.
- Multiplicar el valor de la tasa de interés por la cantidad de recursos. Hacer esto para cada entidad financiera y sumar todos los resultados obtenidos.
- Dividir la suma obtenida en el punto anterior entre el total de los recursos depositados en CDT a 90 días en todas las entidades financieras. Este valor corresponde a la DTF que se utilizará en la siguiente semana.

Para el cálculo de la DTF se puede usar un promedio ponderado lo cual es una ventaja[3].

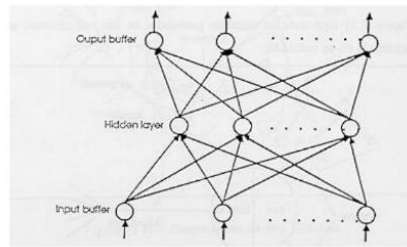
2.2 Redes Neuronales Artificiales

Las Redes Neuronales Artificiales, ANN (Artificial Neural Networks) están inspiradas en las redes neuronales biológicas del cerebro humano. La idea es trabajar de forma parecida a cómo trabaja el cerebro humano. Buscará aprender de la experiencia, generalizar de ejemplos previos a ejemplos nuevos y abstraer las características principales de una serie de datos.

Una red neuronal consistirá en un conjunto de unidades elementales PE conectadas de una forma concreta. El interés residirá no solo en el modelo del elemento PE sino en las formas en que se conectan estos elementos. Existen dos capas con conexiones con el mundo exterior: una capa de entrada, buffer de entrada, donde se presentan los datos a la red, y una capa buffer de salida que mantiene la respuesta de la red a una entrada.

Las demás capas reciben el nombre de capas ocultas. La Figura 1 muestra el aspecto de una Red Neuronal Artificial[4].

Fig. 1. Estructura de una red neuronal simple



3 Metodología

A continuación, se muestra el proceso realizado por medio de la metodología CRISP-DM[5], comenzando por el entendimiento de la base, preparación de los datos, transformación de las variables, selección de las características más importantes para el modelo, aplicación de Redes Neuronales Artificiales (Modelamiento) y la evaluación del método de clasificación por medio de las métricas de comparación para entender la precisión del modelo, para predecir una suscripción a un CDT. Al final el proceso se repetirá hasta tener el objetivo planteado.

3.1 Suscripción de los clientes

Para llevar a cabo el desarrollo del modelo se usó como insumo la base de datos pública del UCI Machine Learning Repository con el nombre de “Bank Marketing Data Set”, la cual cuenta con 17 atributos dentro de ellos la variable de interés para el estudio llamada: “y” – Si el cliente se suscribe al depósito a término, esta es binaria y tiene como respuesta “yes”, o “no”. La base consta de 45211 registros, de los cuales 5289 fueron clientes que adquirieron el CDT, esto nos da una tasa de colocación de 11.7%.

Dentro del set de características se cuenta con información sociodemográfica, duración de la llamada, información financiera del cliente, temporalidad de contacto y frecuencia de la llamada. De esta base se tomará una muestra de entrenamiento y una muestra de testeo de manera aleatoria y se empezará una serie de análisis para determinar qué tan balanceada esta la base, y de ser necesario generar técnicas adecuadas de balanceo que permitan mejorar los niveles de predicción. Una de las variables tiene una alta correlación con la variable respuesta del modelo (duration), y para evitar un sobreajuste se elimina la variable.

El set de características se compone de variables numéricas y categóricas. Para efectos del modelo de Redes a desarrollar, se consideran las variables categóricas en dos grupos inicialmente para su transformación: las binarias en un primer grupo y las de múltiple categoría en un segundo grupo. Para las primeras se tienen: Default, housing,

loan. Para el segundo grupo están: Job, marital, contact, poutcome, education, month; estas últimas son variables ordinales, así que se transformaran bajo su orden respectivo.

3.2 Procesamiento de los datos

Después de entender la estructura de la base, se transforman las variables categóricas mencionadas anteriormente, mediante el algoritmo `get_dummies`, (función de Python, librería Numpy), el cual se encarga de generar columnas con los valores de 0 y 1, siendo 1 la ocurrencia de la variable, transformando la característica original en una serie de variables dummies. Para variables dicotómicas mantiene la misma columna sustituyendo uno de los dos factores por 0 y el otro por 1. Para el caso de las variables ordinales se definirá el orden lógico de las mismas para así tener una transformación secuencial en una única columna donde el primer factor será 0 y este número ascenderá hasta que cada variable en la lista ordenada tenga un numero secuencial asociado.

Del análisis inicial se puede percibir un desbalanceo inicial por parte de la variable respuesta, por ende, se tendrán en cuenta en el desarrollo del proyecto técnicas que den un balance adecuado a la base de entrenamiento y validación. Dentro de las estrategias se contemplan: Subsampling en la clase mayoritaria, Oversampling de la clase minoritaria y resampling con Smote-Tomek, un método de sobre muestreo que genera objetos sintéticos en puntos aleatorios entre objetos de la clase minoritaria y alguno de sus k-vecinos cercanos elegidos al azar. Para el caso particular del Smote-Tomek “genera objetos sintéticos en la clase minoritaria utilizando SMOTE, después se etiquetan como Tomek todos los objetos del conjunto de entrenamiento (minoritario, minoritario sintético y mayoritario) en los cuales su vecino más cercano sea de la clase contraria y, a su vez, ese vecino también tenga como vecino más cercano a un objeto de clase contraria. Una vez etiquetados, los objetos Tomek se eliminan del conjunto de entrenamiento”[6].

Por último, la estandarización de las variables hace que estas lleguen a ser comparables entre ellas y sea más fácil su comprensión. Además de reducir el ruido que pueda tener agregar una escala diferente al modelo a plantear. Para esta etapa se estudiará qué impacto genera el tomar una normalización sobre el total de variables al igual que realizar este procedimiento a un grupo reducido de ellas.

3.3 Implementación de la red neuronal

Una vez realizada la transformación, el balanceo y la estandarización de las variables se comienza a realizar una secuencia de pruebas con el fin de llegar a cumplir el objetivo planteado del 80% de sensibilidad en la prueba. Mediante Colab se ejecutarán en una máquina virtual n1-standard-4 (vCPUs: 4, RAM: 15GB)[7]. El lenguaje elegido para este ejercicio será el de Python, y la función principal para dicha ejecución será `MLPClassifier`, de la librería `sklearn` documentación presentada por `scikit-learn`[8][9].

La base será trabajada sobre una partición con una semilla con el valor 1 y una partición del 70% para entrenamiento y 30% para validación.

3.4 Protocolo experimental

Se genera una primera corrida de los datos sobre la base transformada de un dataframe a un arreglo dado que los modelos de Machine Learning no reconocen ese formato. Teniendo como referencia el primer modelo se plantea una ejecución con 2 capas, 2 neuronas en la primera y 1 en la segunda capa bajo el algoritmo lbfgs; este primer modelo revela los problemas de trabajar con los datos con las mínimas transformaciones (conversión de variables categóricas a numéricas), bajo su entrenamiento predice que solo dos clientes tomaran la campaña con una sensibilidad de 0.063%.

Post el primer entrenamiento se realizan una serie de ejercicios que prueban los tres métodos de balanceo de los clientes. Para un balanceo exitoso se tomará la base completa y sobre esta se trabajará el método seleccionado. Mantenido los parámetros de la red neuronal fija se tienen mejores resultados de sensibilidad.

Fig. 2. Resultado de calentamiento modelo entrenamiento

```
-----
Valor      Predicho
-----
-          no      yes
Valor      no      2353    9624
Verdadero  yes      1106    10871
SCORE entrenamiento: 0.5548577563070317
Error en entrenamiento: 0.4451422436929683
SCORE prueba: 0.5520581113801453
Error en prueba: 0.44794188861985473
Sensibilidad: 0.907656341320865
```

Como se observa en el resultado a pesar de tener una sensibilidad superior a los 80% planteados no se tiene una buena predicción del modelo sobre esta metodología, por ende, se procede a realizar tanto la estandarización, como el balanceo de la base, y con ello se empezará a buscar qué hiperparámetros pueden llegar a tener los mejores resultados.

Los parámetros que se varían los siguientes:

- Solver: ['sgd', 'adam', 'lbfgs'],
- Activation: ['logistic', 'relu', 'tanh', 'identity']
- Hidden_layer_sizes: [(10), (10, 12), (10,12,14), (10,10,10)]
- Alpha: [0.001, 1, 1000]
- Learning_rate_init: [0.001, 0.01, 0.1]

Por la cantidad de iteraciones definidas se limita el número de combinaciones a probar (esto está relacionado al tiempo de máquina). Para este caso se tomarán 50 iteraciones sobre 5 combinaciones aleatorias, con una semilla de 1 en ambos casos. Los resultados obtenidos son los siguientes:

(param_activation, V1), (param_alpha, V2), (param_hidden_layer_sizes, V3), (param_solver, V4), (mean_test_score, V5), (std_test_score, V6), (rank_test_score, V7), (mean_train_score, V8), (std_train_score, V9)

Table 1. resultados de hiperparametros sobre muestra estandarizada y balanceada

V1	V2	V3	V4	V5	V6	V7	V8	V9
relu	1	(10,12)	sgd	0.7066	0.0115	6.0000	0.7214	0.0044
relu	1	(10,12)	adam	0.7072	0.0101	4.0000	0.7365	0.0024
relu	1	(10,12,14)	sgd	0.7044	0.0085	8.0000	0.7275	0.0015
relu	1	(10,12,14)	adam	0.7063	0.0099	7.0000	0.7361	0.0021
relu	0.01	(10,12)	sgd	0.6957	0.0101	16.0000	0.7197	0.0035
relu	0.01	(10,12)	adam	0.6969	0.0100	14.0000	0.7442	0.0034
relu	0.01	(10,12,14)	sgd	0.6958	0.0089	15.0000	0.7257	0.0043
relu	0.01	(10,12,14)	adam	0.6981	0.0100	12.0000	0.7694	0.0068
tanh	1	(10,12)	sgd	0.7001	0.0047	11.0000	0.7080	0.0023
tanh	1	(10,12)	adam	0.7116	0.0118	1.0000	0.7252	0.0036
tanh	1	(10,12,14)	sgd	0.7074	0.0094	3.0000	0.7233	0.0050
tanh	1	(10,12,14)	adam	0.7104	0.0100	2.0000	0.7351	0.0059
tanh	0.01	(10,12)	sgd	0.6971	0.0072	13.0000	0.7142	0.0028
tanh	0.01	(10,12)	adam	0.7026	0.0137	9.0000	0.7803	0.0045
tanh	0.01	(10,12,14)	sgd	0.7069	0.0083	5.0000	0.7295	0.0054
tanh	0.01	(10,12,14)	adam	0.7022	0.0042	10.0000	0.7992	0.0069

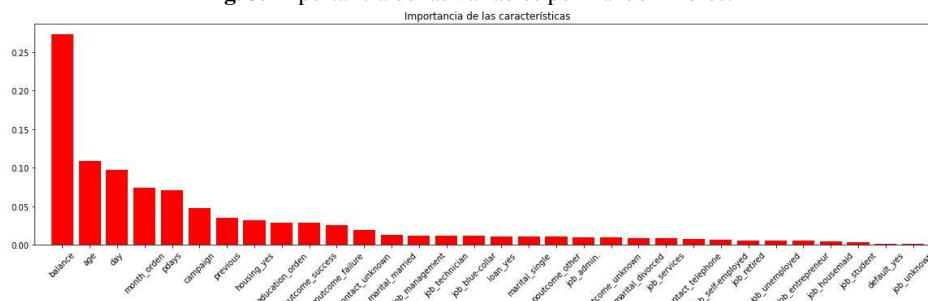
- Solver: ['adam'],
- Activation: ['tanh']
- Hidden_layer_sizes: [(10, 12)]
- Alpha: 1

Sobre esta muestra y las pruebas adicionales se observa una cota en la sensibilidad del 72%. Al ver que el modelo no sube de ese valor se hace un ajuste final del modelo.

3.5 Ajuste final

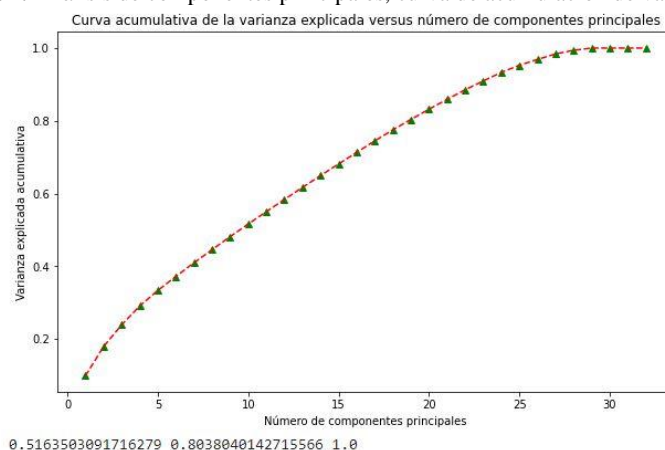
Teniendo los hiperparámetros la idea es reducir el número de variables que puedan estar afectando la predicción del modelo, se usará el algoritmo de random forest[10]. Con este algoritmo se analizarán las variables más importantes para el modelo. En la siguiente grafica se muestran las que resultaron más importantes:

Fig. 3. Importancia de las variables por Random Forest



Adicional a este análisis se complementa el uso de componentes principales para ver con que numero de variables se tiene la mayor cantidad de varianza recogida por el modelo.

Fig. 4. Análisis de componentes principales, curva de acumulación de varianza



Con estas variables y con los parámetros encontrados anteriormente se genera el modelo final que alcanza más del 80% de sensibilidad planteado.

4 Resultados

- **Iteración 1:** ejecución del modelo con características completas sin estandarizar y con datos desbalanceados.
- **Iteración 2:** ejecución del modelo con características completas sin estandarizar y con datos balanceados por submuestreo en clase mayoritaria con algoritmo vecino más cercano sobre conjunto de datos de entrenamiento.
- **Iteración 3:** ejecución del modelo con características completas sin estandarizar y con datos balanceados por submuestreo en clase mayoritaria con algoritmo vecino más cercano sobre conjunto de datos completos.

- **Iteración 4:** ejecución del modelo con características completas sin estandarizar y con datos balanceados por sobre muestreo en clase minoritaria.
- **Iteración 5:** ejecución del modelo con características completas sin estandarizar y con datos balanceados mediante algoritmo Smote-Tomek.
- **Iteración 6:** ejecución del modelo con características completas aplicando estandarización sobre todas las variables del conjunto de datos de la iteración 4.
- **Iteración 7:** ejecución del modelo con características completas aplicando estandarización sobre las variables ordinales del conjunto de datos de la iteración 4.
- **Iteración 8:** ejecución del modelo con características completas variando hiperparámetros mediante algoritmo de búsqueda aleatoria sobre la estandarización de los datos de la iteración 4. Se ejecutan 50 combinaciones de hiperparámetros y selecciona el mejor.
- **Iteración 9:** ejecución del modelo con características completas variando hiperparámetros mediante algoritmo de búsqueda exhaustiva sobre la estandarización de los datos de la iteración 4. Se ejecutan 16 combinaciones de los mejores hiperparámetros obtenidos en la iteración 8.
- **Iteración 10:** ejecución del modelo seleccionando las características más relevantes del conjunto de datos estandarizados utilizando los mejores hiperparámetros obtenidos de la iteración 9.
- **Iteración 11:** ejecución del modelo seleccionando las características más relevantes del conjunto de datos estandarizados con los mejores hiperparámetros obtenidos de la iteración 9 y con datos balanceados mediante algoritmo Smote-Tomek.
- **Iteración 12:** ejecución del modelo seleccionando con características completas estandarizadas con los mejores hiperparámetros y con datos balanceados mediante algoritmo Smote-Tomek.

11976	1	99,99%
88,29%	0,01%	0,01%
1586	1	0,06%
11,69%	0,01%	99,94%
88,31%	50,00%	88,30%
11,69%	50,00%	11,70%
Precisión		88,2999%
Sensibilidad		0,0630%

Fig 5. Matriz de Confusión Iteración 1

4062	7915	33,92%
29,95%	58,35%	66,08%
857	730	46,00%
6,32%	5,38%	54,00%
82,58%	8,44%	35,33%
17,42%	91,56%	64,67%
Precisión		35,3288%
Sensibilidad		45,9987%

Fig 6. Matriz de Confusión Iteración 2

1299	288	81,85%
40,93%	9,07%	18,15%
800	787	49,59%
25,20%	24,80%	50,41%
61,89%	73,21%	65,72%

2353	9624	19,65%
9,82%	40,18%	80,35%
1106	10871	90,77%
4,62%	45,38%	9,23%
68,03%	53,04%	55,21%

38,11%	26,79%	34,28%
Precisión		65,7215%
Sensibilidad		49,5904%

Fig 7. Matriz de Confusión Iteración 3

31,97%	46,96%	44,79%
Precisión		55,2058%
Sensibilidad		90,7656%

Fig 8. Matriz de Confusión Iteración 4

7703 33,00%	3967 17,00%	66,01% 33,99%
5112 21,90%	6558 28,10%	56,20% 43,80%
60,11% 39,89%	62,31% 37,69%	61,10% 38,90%
Precisión		61,1011%
Sensibilidad		56,1954%

Fig 9. Matriz de Confusión Iteración 5

1232 38,82%	355 11,18%	77,63% 22,37%
499 15,72%	1088 34,28%	68,56% 31,44%
71,17% 28,83%	75,40% 24,60%	73,09% 26,91%
Precisión		73,0939%
Sensibilidad		68,5570%

Fig 10. Matriz de Confusión Iteración 6

1320 41,59%	267 8,41%	83,18% 16,82%
887 27,95%	700 22,05%	44,11% 55,89%
59,81% 40,19%	72,39% 27,61%	63,64% 36,36%
Precisión		63,6421%
Sensibilidad		44,1084%

Fig 11. Matriz de Confusión Iteración 7

1189 37,46%	398 12,54%	74,92% 25,08%
443 13,96%	1144 36,04%	72,09% 27,91%
72,86% 27,14%	74,19% 25,81%	73,50% 26,50%
Precisión		73,5035%
Sensibilidad		72,0857%

Fig 12. Matriz de Confusión Iteración 8

753 35,59%	305 14,41%	71,17% 28,83%
295 13,94%	763 36,06%	72,12% 27,88%
71,85% 28,15%	71,44% 28,56%	71,64% 28,36%
Precisión		71,6446%
Sensibilidad		72,1172%

Fig 13. Matriz de Confusión Iteración 9

1095 34,50%	492 15,50%	69,00% 31,00%
567 17,86%	1020 32,14%	64,27% 35,73%
65,88% 34,12%	67,46% 32,54%	66,64% 33,36%
Precisión		66,6352%
Sensibilidad		64,2722%

Fig 14. Matriz de Confusión Iteración 10

10233	1416	87,84%
43,92%	6,08%	12,16%
2508	9140	78,47%
10,77%	39,23%	21,53%
80,32%	86,59%	83,16%
19,68%	13,41%	16,84%
Precisión		83,1566%
Sensibilidad		78,4684%

Fig 15. Matriz de Confusión Iteración 11

10621	1049	91,01%
45,51%	4,49%	8,99%
2081	9589	82,17%
8,92%	41,08%	17,83%
83,62%	90,14%	86,59%
16,38%	9,86%	13,41%
Precisión		86,5895%
Sensibilidad		82,1680%

Fig 16. Matriz de Confusión Iteración 12

Seleccionamos como modelo final el obtenido en la iteración 12 al que le realizamos una prueba de validación cruzada de K iteraciones con 50 muestras, para las que se obtiene un promedio de precisión de 86.66% y un promedio de sensibilidad de 60.9830%. Cuando se prueba el modelo con los datos de prueba originales se obtiene la siguiente matriz de confusión y curva ROC:

9910	1760	84,92%
42,46%	7,54%	15,08%
2185	9485	81,28%
9,36%	40,64%	18,72%
81,93%	84,35%	83,10%
18,07%	15,65%	16,90%
Precisión		83,0977%
Sensibilidad		81,2768%

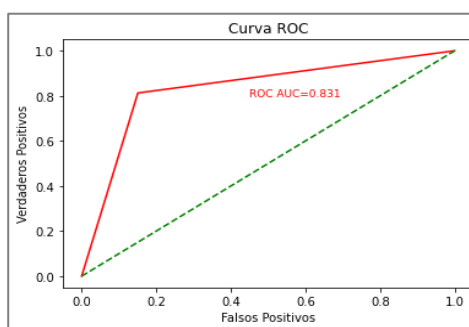


Fig 17. Matriz de Confusión y Curva ROC modelo final con datos de prueba primarios

5 Conclusiones

- Logramos indicadores de precisión y sensibilidad superiores a los artículos revisados en la bibliografía.
- Las redes neuronales artificiales logran clasificar de manera efectiva el perfil de los clientes y así aumentar la probabilidad de aceptación del producto o servicio al utilizar una campaña de marketing directo.

- Un análisis previo de las variables para poder identificar posibles problemas que generen sobre ajuste como lo son variables altamente correlacionadas con la variable objetivo o un desbalanceo.
- Es necesario probar varias técnicas diferentes en cada etapa del desarrollo, a veces los resultados suelen favorecer una metodología inicialmente, pero al llevar a la evaluación de los resultados no presenta el mejor resultado.
- Cuando se llegan a mínimos locales y no se tiene un incremento es los estadísticos evaluación se debe replantear la cantidad de variables y el ruido que estas puedan estar generando en el modelo.
- El método de Smote - Tomek, por ser una técnica que balancea tanto la cantidad de objetos de interés como los de no interés suele tener mejores resultados en este tipo de datos. No obstante, no es bueno trabajar solo con una técnica inicialmente.
- Una buena forma de obtener mejores resultados puede ser la reducción de variables en un modelo evaluado.

6 Bibliografía

- [1] “Invertir en un CDT le puede dejar una rentabilidad hasta de 5,50% según el plazo.” <https://www.larepublica.co/finanzas/invertir-en-un-cdt-le-puede-dejar-una-rentabilidad-hasta-de-550-segun-el-plazo-3220582> (accessed Mar. 20, 2022).
- [2] “UCI Machine Learning Repository: Bank Marketing Data Set.” <https://archive.ics.uci.edu/ml/datasets/bank+marketing> (accessed Mar. 12, 2022).
- [3] “Depósitos a término fijo - Enciclopedia | Banrepcultural.” https://enciclopedia.banrepcultural.org/index.php/Dep%C3%B3sitos_a_t%C3%A9rmino_fijo (accessed Mar. 20, 2022).
- [4] X. B. Olabe, “REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES.”
- [5] “Conceptos básicos de ayuda de CRISP-DM - Documentación de IBM.” <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=dm-crisp-help-overview> (accessed Mar. 20, 2022).
- [6] J. Ariel, C. Ochoa, J. Francisco, and M. Trinidad, “SMOTE-D, UNA VERSIÓN DETERMINISTA DE SMOTE.”
- [7] “Acerca de las familias de máquinas | Documentación de Compute Engine | Google Cloud.” <https://cloud.google.com/compute/docs/machine-types> (accessed Mar. 20, 2022).
- [8] “sklearn.neural_network.MLPClassifier — scikit-learn 1.0.2 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (accessed Mar. 20, 2022).
- [9] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

- [10] “sklearn.ensemble.RandomForestClassifier — scikit-learn 1.0.2 documentation.” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> (accessed Mar. 20, 2022).
- [11] Guido, G., Prete, M. I., Miraglia, S., & De Mare, I. (2011). Targeting direct marketing campaigns by neural networks. *Journal of Marketing Management*, 27(9-10), 992-1006.
- [12] Elsalamony, H. A., & Elsayad, A. M. (2013). Bank direct marketing based on neural network and C5.0 Models. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(6).
- [13] Al-Shayea, Q. K. (2013). Evaluating marketing campaigns of banking using neural networks. In *Proceedings of the World Congress on Engineering* (Vol. 2).
- [14] Liu, H. H., & Ong, C. S. (2008). Variable selection in clustering for marketing segmentation using genetic algorithms. *Expert Systems with Applications*, 34(1), 502-510.
- [15] Khan, N., & Khan, F. (2013). Fuzzy based decision making for promotional marketing campaigns. *International Journal of Fuzzy Logic Systems*, 3(1), 64-77.
- [16] Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62, 22-31.