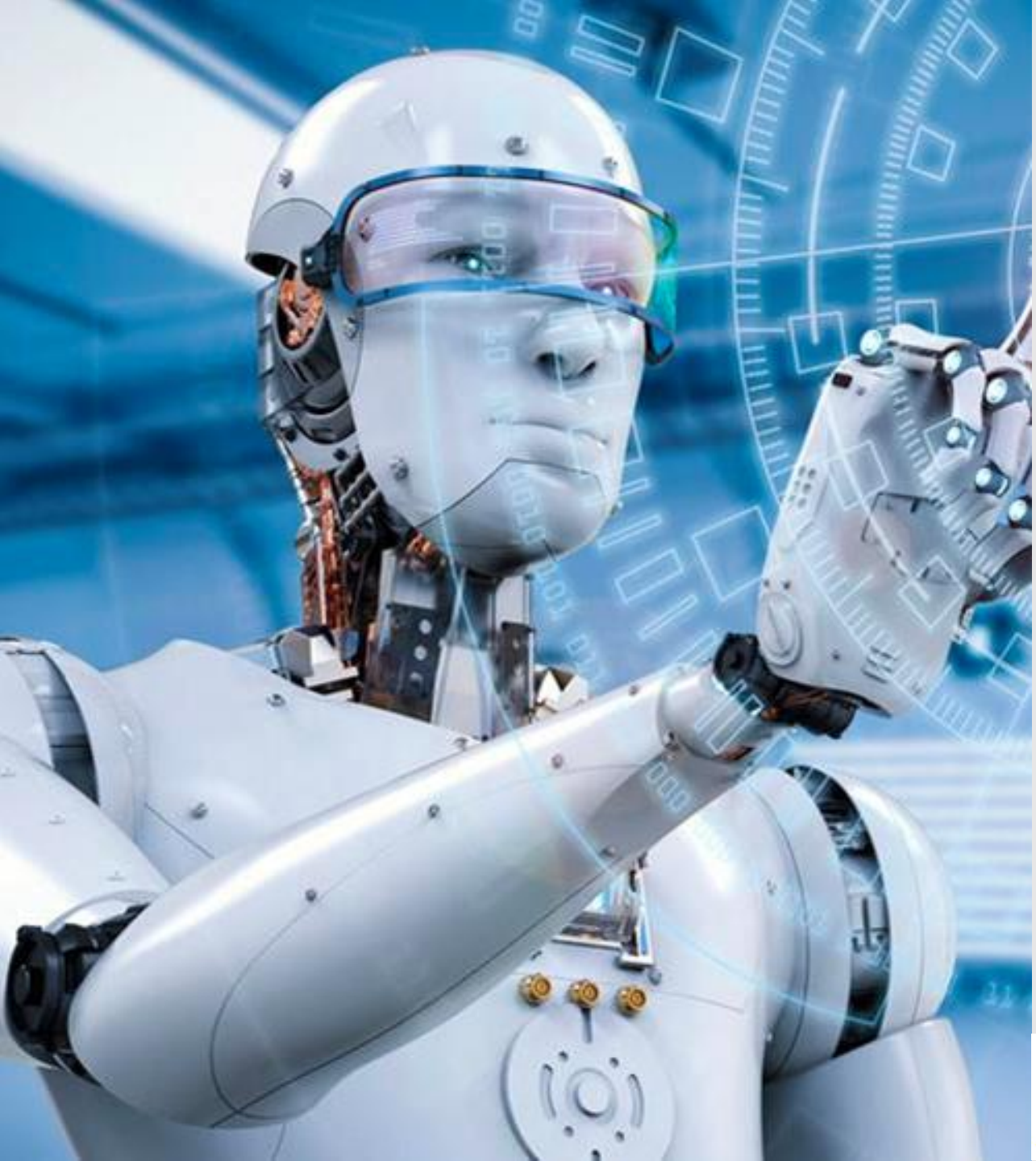


Optimización de colocación en campañas telefónicas por medio de sistemas inteligentes

SISTEMAS INTELIGENTES

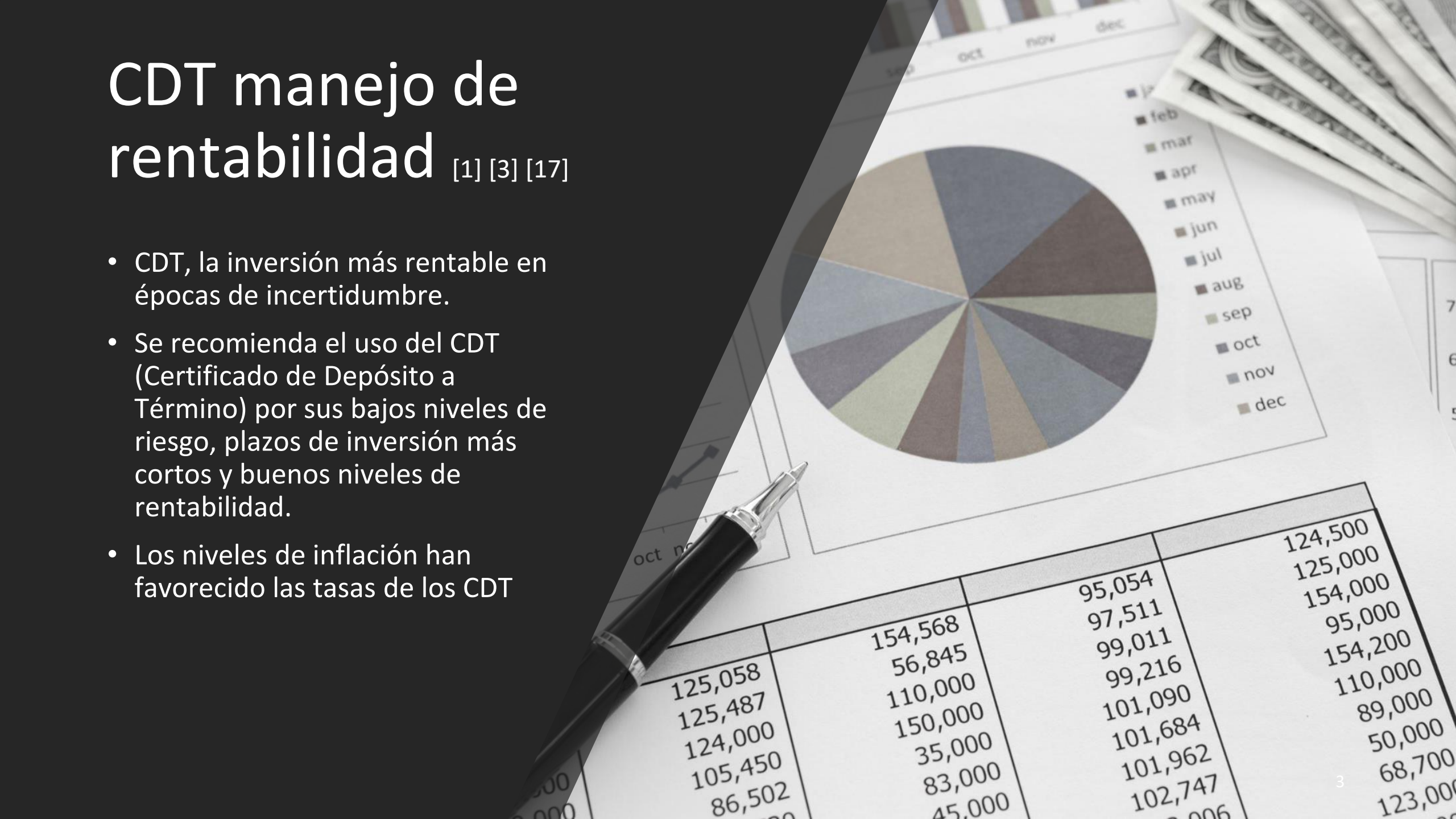
- *Edison Leonardo Neira Espitia*
- *Sergio Andrés Rairán*
- *Sebastián Herrera Monterrosa*

Agenda

- *Contextualización*
- *Estado del Arte*
- *Solución del Problema*
- *Herramientas de Implementación*
- *Protocolo Experimental*
- *Resultados*
- *Conclusiones*

CDT manejo de rentabilidad [1] [3] [17]

- CDT, la inversión más rentable en épocas de incertidumbre.
- Se recomienda el uso del CDT (Certificado de Depósito a Término) por sus bajos niveles de riesgo, plazos de inversión más cortos y buenos niveles de rentabilidad.
- Los niveles de inflación han favorecido las tasas de los CDT



Base de datos para investigación[2]

- UCI Machine Learning Repository - “Bank Marketing Data Set”
- **https://archive.ics.uci.edu/ml/datasets/bank+marketing**

Base de datos para investigación

45,211 clientes con aplicación de la campaña telefónica

16 variables analizadas

100% completitud de la base

Artículo 1 – Artículo científico[16]

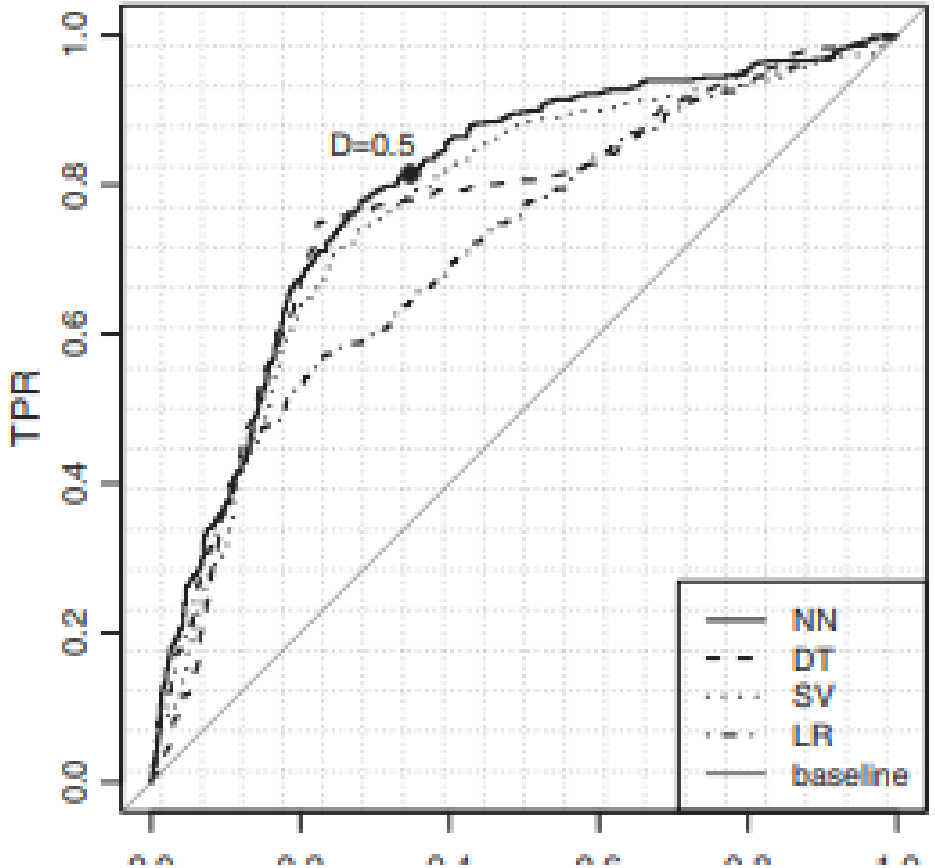
A data-driven approach to predict the success of bank telemarketing

Sérgio Moro [a,*], Paulo Cortez [b], Paulo Rita [a]

[a] ISCTE-IUL, Business Research Unit (BRU-IUL), Lisboa, Portugal
[b] ALGORITMI Research Centre, Univ. of Minho, 4800-058 Guimarães, Portugal

Puntos clave

Trabajan los algoritmos de árboles de decisión, regresión logística, support-vector machines y redes neuronales artificiales. Concluyen que las redes neuronales son la mejor técnica obteniendo un 74,78% de precisión y un 65,29% de sensibilidad.



NN model
(ROC point for D=0.5):

	Predicted	
Target	failure	success
failure	617 (48%)	140 (11%)
success	186 (14%)	350 (27%)

Artículo 2 – Artículo científico[13]

Evaluating Marketing Campaigns of Banking Using Neural Networks

Qeethara Kadhim Al-Shayea, *Member, IAENG*

Puntos clave

Trabajan el problema mediante las redes de propagación retroalimentadas y las redes neuronales artificiales.

Redes neuronales artificiales

28 2,75%	**17** 1,67%	62,22% 37,78%
93 9,13%	**881** 86,46%	90,45% 9,55%
23,14% 76,86%	98,11% 1,89%	89,21% 10,79%
Precisión		**89,2051%**
Sensibilidad		**62,2222%**

Redes de propagación retroalimentadas

90 2,57%	**62** 1,77%	59,21% 40,79%
310 8,86%	**3037** 86,80%	90,74% 9,26%
22,50% 77,50%	98,00% 2,00%	89,37% 10,63%
Precisión		**89,3684%**
Sensibilidad		**59,2105%**

Artículo 3 – Artículo científico[18]

Evaluation of Classification and Ensemble Algorithms for Bank Customer Marketing Response Prediction

Olatunji Apampa
Tilburg University
The Netherlands

ABSTRACT

This article attempts to improve the performance of classification algorithms used in the bank customer marketing response prediction of an unnamed Portuguese bank using the Random Forest ensemble.

Puntos clave

- Se encontró que el algoritmo del árbol de decisiones (CART) se desempeñó mejor que los algoritmos de regresión logística (LR) y Naïve Bayes (NB) y arrojó un área bajo la curva (AUC) y un valor de precisión de clasificación (CA) de 76.6 %.

- Balancear los datos proporciona mejores resultados.

Artículo 4 – Artículo científico[19]

Journal of Engineering and Natural Sciences
Mühendislik ve Fen Bilimleri Dergisi

Sigma 32, 142-152, 2014

Research Article / Araştırma Makalesi

COMPARISON OF DATA MINING TECHNIQUES FOR DIRECT MARKETING CAMPAINGS

Esra AKDENİZ DURAN[*1], Ayça PAMUKCU[2], Hazal BOZKURT[2]

[1]İstanbul Medeniyet University, Faculty of Sciences, Department of Statistics, Üsküdar-İSTANBUL
[2]Marmara University, Faculty of Arts and Sciences, Department of Statistics, Göztepe-İSTANBUL

Received/Geliş: 27.07.2013 Revised/Düzeltme: 19.11.2013 Accepted/Kabul: 05.02.2014

Puntos clave

- El éxito de los bancos depende del uso eficaz de las estrategias de marketing directo.

- El modelo de red neuronal artificial tiene una precisión de clasificación del 90.8 % y una sensibilidad de 41.3% tanto para el train como para los conjuntos de datos de validación test.

Artículo 5 – Artículo científico[12]

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958, Volume-2, Issue-6, August 2013

Bank Direct Marketing Based on Neural Network

Hany. A. Elsalamony, Alaa. M. Elsayad

Puntos clave

Utilizan los mecanismos de redes neuronales de perceptrón multicapa y modelo de árbol de decisiones de Ross Quinlan, obteniendo los siguientes resultados.

Model	Partition	Accuracy	Sensitivity
MLPNN	Training	90.84%	63.64%
	Testing	90.32%	60.12%
C5.0	Training	93.23%	76.75%
	Testing	90.09%	59.06%

Artículo 6 – Artículo científico[11]

Targeting direct marketing campaigns by neural networks

Gianluigi Guido, *University of Salento, Italy*
M. Irene Prete, *University of Salento, Italy*
Stefano Miraglia, *Imperial College London, UK*
Irma De Mare, *Leroy Merlin, Italy*

Puntos clave

- Abordan el problema utilizando técnicas de aprendizaje de máquina como regresión múltiple y regresión logística, y también se desarrolla usando redes neuronales.

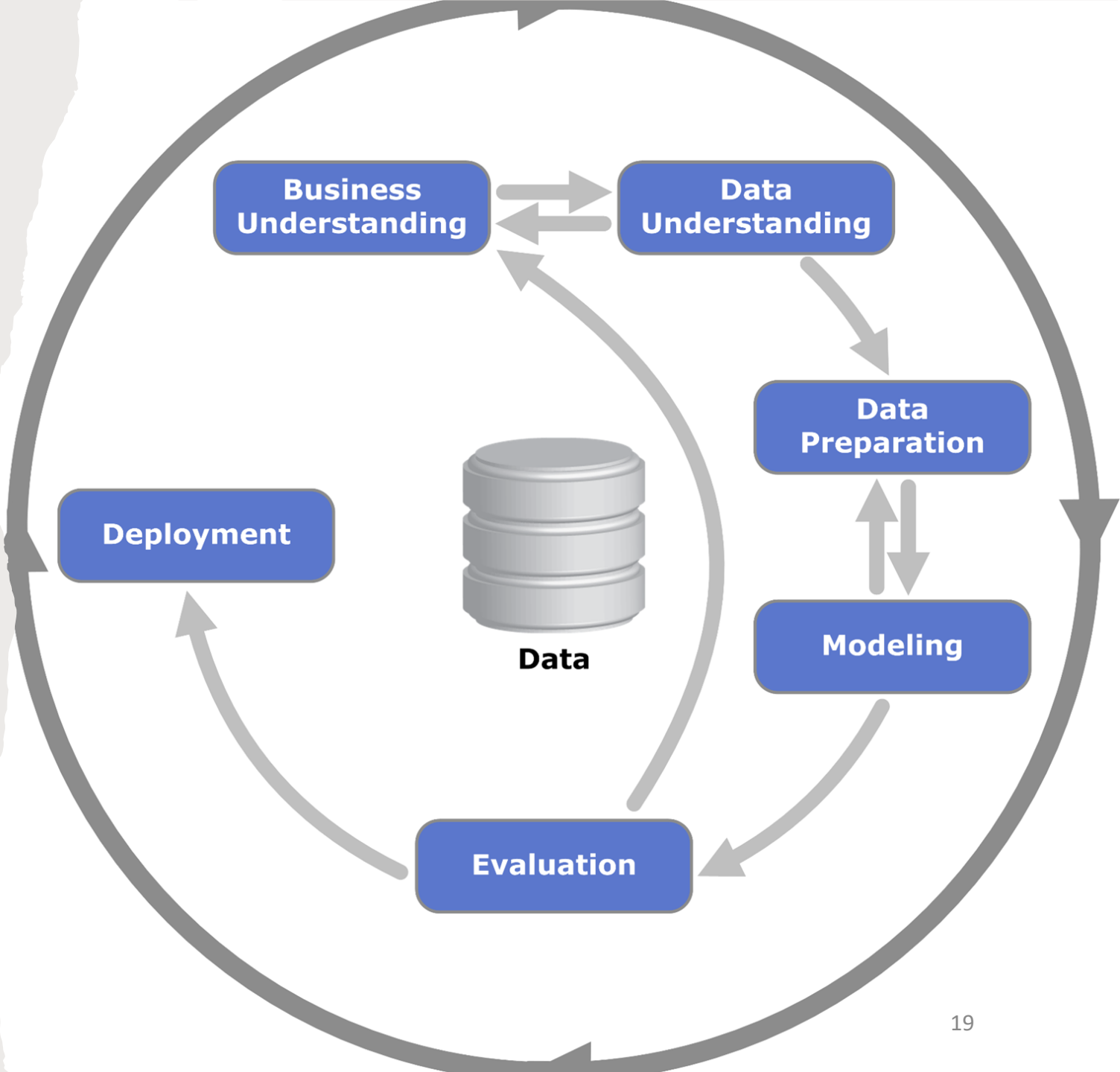
- Concluyen que las redes neuronales son una mejor técnica para el problema como una mejor técnica para este problema.

Solución del problema

- Metodología

Variable respuesta

- La respuesta se define: toma o no la campaña ofrecida vía telefónica.
- Se manejan variables socio-demográficas, comportamiento financiero y estado de la llamada.
- Las variables caracterizan la respuesta.
- Se realizarán los cambios y transformaciones respectivas a la base.



CRISP-DM[5]

Proceso de solución

Tratamiento

- Conversión de características

Balanceo

- Aplicación técnicas

Estandarización

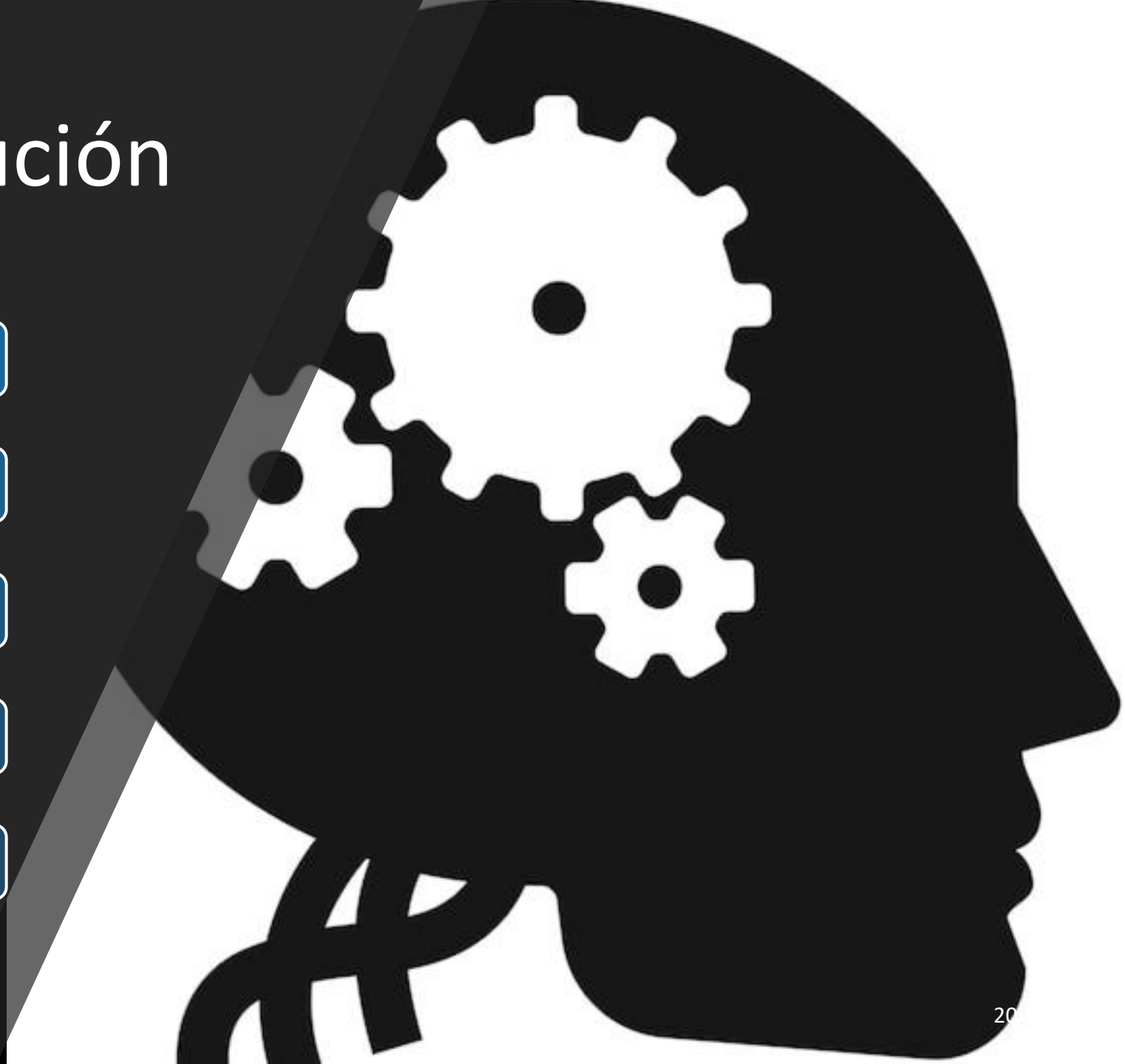
- Transformación de variables

Entrenamiento RNA

- Pruebas del modelo

Medición resultados

- Cálculo de estadísticos



Tratamiento

- Por medio de la función get_dummies, se transforman variables categóricas a numéricas.

```
[ ]  df_piv=pd.get_dummies(df_piv,columns=['job','marital','poutcome'])
```

```
[ ]  df_piv=pd.get_dummies(df_piv,columns=['default','housing','loan','contact','y'],drop_first = True)
```

- La base se convierte en un arreglo

- La variable respuesta es expresada en términos de 0 y 1

Tratamiento de característica

De acuerdo al conjunto de datos y posterior al análisis de características aplicamos las siguientes estrategias a cada conjunto de variables:

Variables Nominales

- **Job**
- **Marital**
- **Default**
- **Housing**
- **Loan**
- **Contact**
- **Poutcome**
- **Y (Target)**

Estrategia usada

pd.get_dummies

Variables Ordinales

- **Education**
- **Month**

Estrategia usada

OrdinalEncoder

Variables Descartadas

- **Duration**

Estrategia usada

Eliminar: Variable que tiene una alta relación con la campaña por lo cual nos puede sobre ajustar el modelo, pues indica tiempo de duración de la llamada y es 0 cuando no ha sido contactado

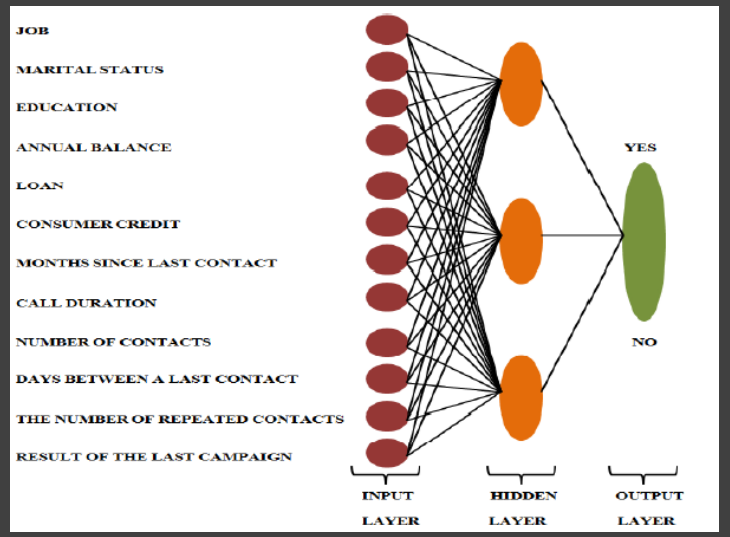
Variables Estandarizadas

- **age**
- **balance**
- **day**
- **campaign**
- **pdays**
- **previous**

Estrategia usada

StandardScaler()

Redes Neuronales Artificiales



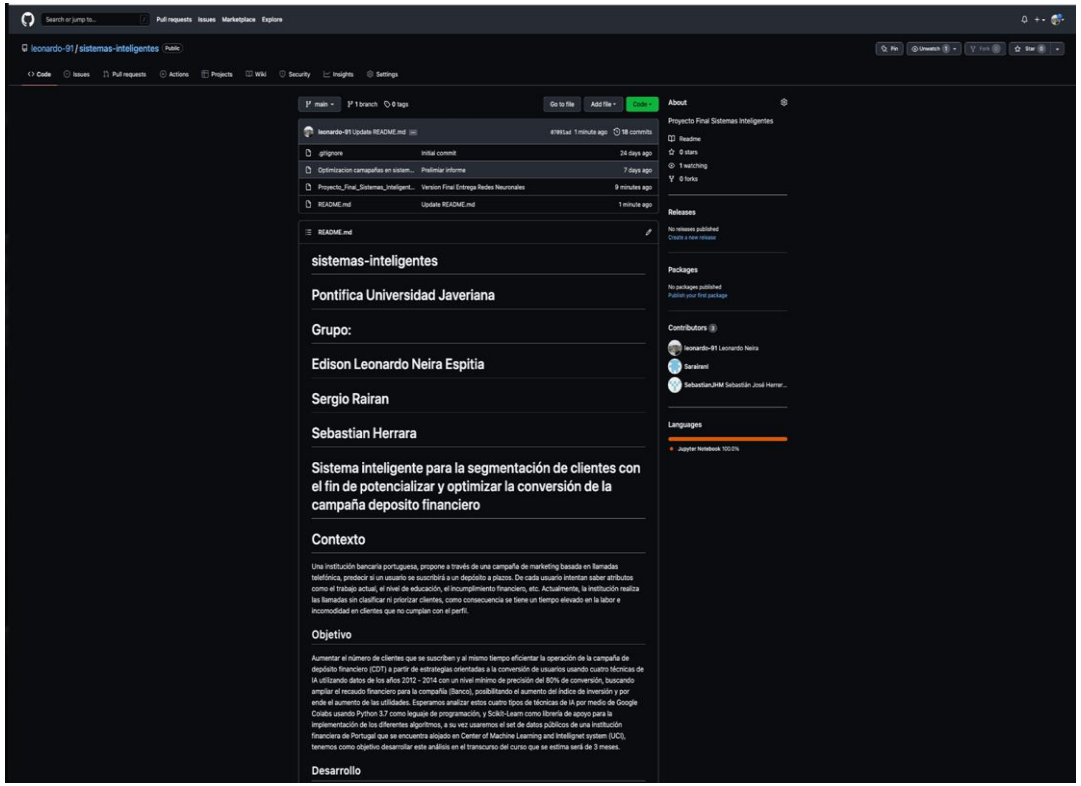
- Se clasifican las personas que adquieren la campaña
- Hay un aprendizaje con cada iteración
- Se balancean los datos de entrada

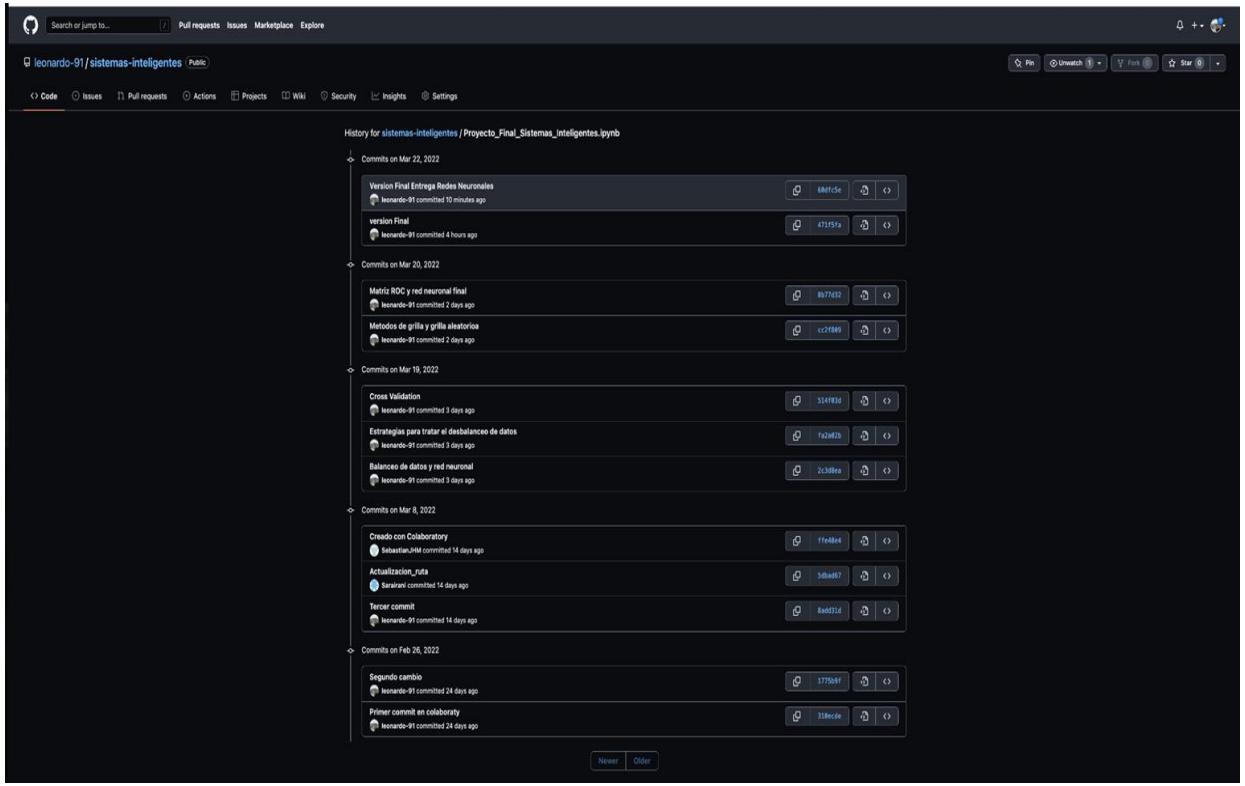
- Se realiza la búsqueda de los mejores hiper parámetros
- El algoritmo se calibra con los ajustes de la base

Software Matemático

- Python como lenguaje de programación
- Google Collabs como entorno de desarrollo
- Scikit Learn para implementación de redes neuronales
- Librerías Pandas y Numpy para manejo de DataFrame
- Matplotlib para graficar indicadores y resultados

Repositorio GitHub





https://github.com/leonardo-91/sistemas-inteligentes

Protocolo Experimental

En nuestra protocolo experimental empezamos la ejecución sin hacer ningún tratamiento adicional a las variables aparte de la conversión de características usando mayormente los hiper-parámetros por default (distribución 70% Tr -- 30% Ts):

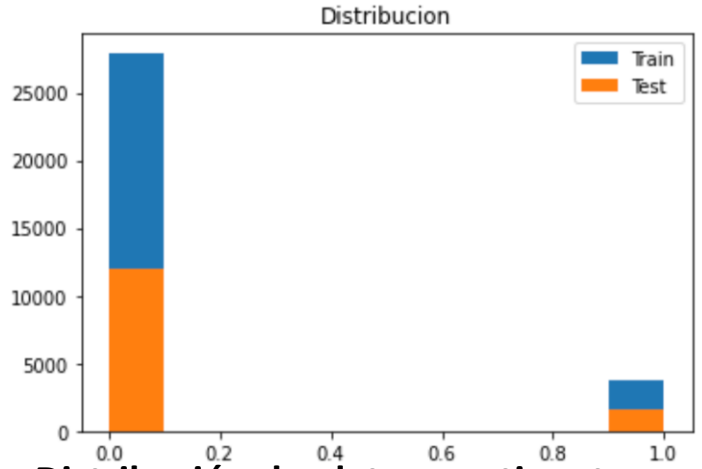
	Hiperparametros						
Cantidad Variables	Solver	max-iter	random_state	hidden_layer_sizes	alpha	activation	learning_rate_init
32	lbfgs	100	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)

Salida	
Accuracy train	0.8827
accuracy score	0.8829
Sensibilidad	0.0006

F1 - Parámetros y salidas

		Valor	Predicho
	-	no	yes
Valor	no	11976	1
Verdadero	yes	1586	1

F2 - Matriz de confusión



F3 - Distribución de datos vs etiquetas

En este caso aunque el modelo tiene una muy buena exactitud vemos muy **mala sensibilidad** que es nuestra métrica objetivo, adicionalmente con base a las figuras **F1** y **F3** vemos que el modelo está generalizando y presenta un claro desbalance en las etiquetas **tipo 0** (No tomar la campaña).

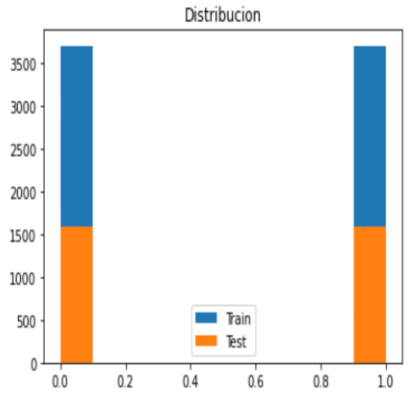
Protocolo Experimental

Dado el desbalance de las clases en nuestro modelo procedemos aplicar 3 técnicas de ML para tratar este caso (distribución 70% Tr -- 30% Ts):

	Hiperparametros						
Cantidad Variables	**Solver**	**max-iter**	**random_state**	**hidden_layer_sizes**	**alpha**	**activation**	**learning_rate_init**
32	lbfgs	200	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)

Undersampling

```
Distribucion Original Counter({0: 27945, 1: 3702})
Distribucion despuest del subsampling Counter({0: 3702, 1: 3702})
```

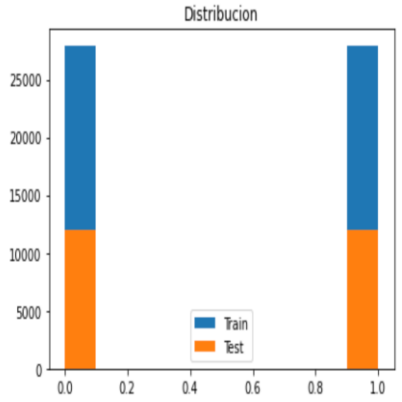


Salida		
Accuracy train	**accuracy score**	**Sensibilidad**
0.6569	**0.6572**	**0.4959**

		Valor	Predicho
	-	no	yes
Valor	no	1299	288
Verdadero	yes	800	787

Oversampling

```
Distribucion Original Counter({0: 39922, 1: 5289})
Distribution despues del oversampling Counter({0: 39922, 1: 39922})
```

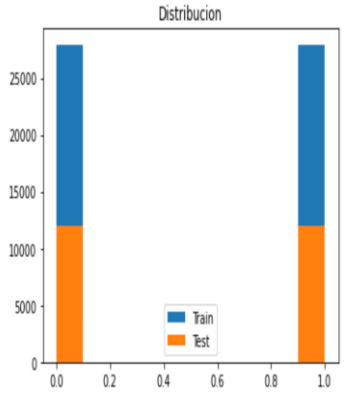


Salida		
Accuracy train	**accuracy score**	**Sensibilidad**
0.5548	**0.552**	**0.9076**

		Valor	Predicho
	-	no	yes
Valor	no	2353	9624
Verdadero	yes	1106	10871

SmoteTomek (Combinada)

```
Distribucion Original Counter({0: 39922, 1: 5289})
Distribution despues smote-Tomek Counter({0: 38900, 1: 38900})
```



Salida		
Accuracy train	**accuracy score**	**Sensibilidad**
0.6128	**0.611**	**0.5619**

		Valor	Predicho
	-	no	yes
Valor	no	7703	3967
Verdadero	yes	5112	6558

Protocolo Experimental

Luego de realizar el tratamiento de desbalanceo de datos decidimos seleccionar **somoteTomek** como mejor estrategia y realizamos un aproximado total de 500 de secciones de ejecución de aprendizaje en las redes neuronales variando cantidad de características e hiperparámetros. Comenzando con un ingenuo y a medida que no se llega al resultado se ajustan los hiperparámetros

Hiperparametros

		Hiperparametros						
Cantidad Variables	Standarizacion	Solver	max-iter	random_state	hidden_layer_sizes	alpha	activation	learning_rate_init
32	No	lbfgs	100	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)
32	No	lbfgs	200	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)
32	No	lbfgs	200	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)
32	No	lbfgs	200	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)
32	SI	lbfgs	200	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)
32	SI (PARCIAL)	lbfgs	200	1	(2,1)	0.0001 (Default)	relu (Default)	0.001 (Default)
32	SI	['sgd', 'adam', 'lbfgs']	5000	1	[(10), (10, 12),(10,12,14),(10,10,10)]	[1.e-03, 1., 1.e+03]	['logistic', 'relu', 'tanh', 'identity']	[0.001, 0.01, 0.1]
32	SI	['sgd', 'adam']	5000	1	[(10,12),(10,12,14)]	[1.e-03, 1.]	['relu', 'tanh']	[0.01, 0.1]

Secciones Individuales

100 Secciones ejecutadas a través del método - RandomizedSearchCV

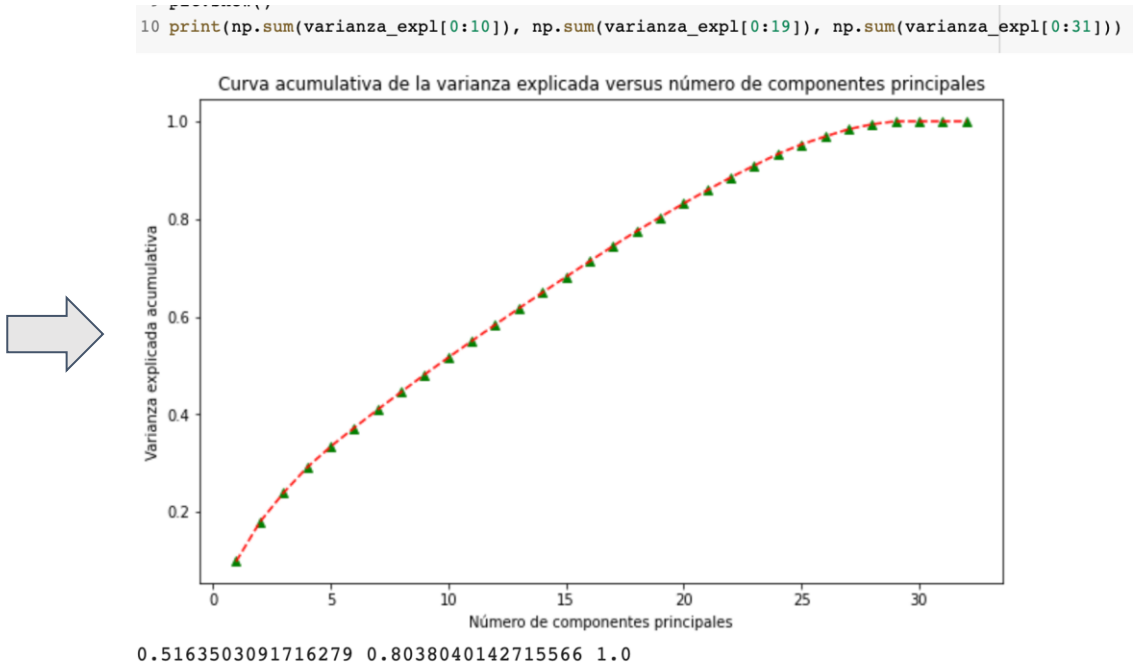
16 Secciones ejecutadas a través del método - GridSearchCV

Protocolo Experimental

Luego de realizar el tratamiento de desbalanceo de datos decidimos seleccionar **Smote-Tomek** como mejor estrategia y realizamos un aproximado total de 500 de secciones de ejecución de aprendizaje en las redes neuronales variando cantidad de características e hiperparametros .

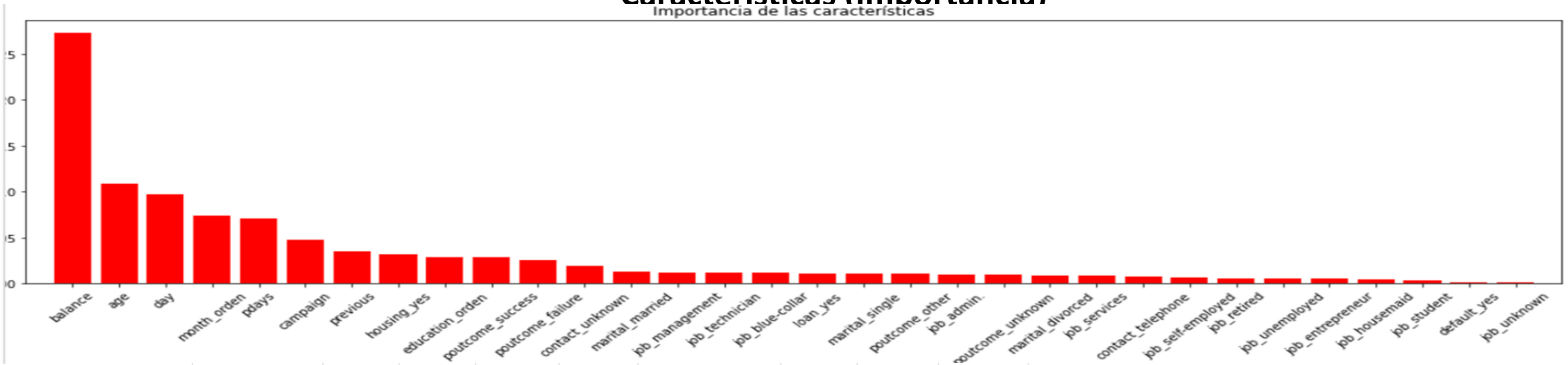
Características (PCA)

		Hiperparametros						
Cantidad Componentes	Standarizacion	Solver	max-iter	random_state	hidden_layer_sizes	alpha	activation	learning_rate_init
2	SI	sgd	2000	1	(10,12)	1	relu	0.01
2	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
4	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
6	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
8	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
10	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
12	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
14	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
16	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
18	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
20	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
22	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
24	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
26	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
28	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
30	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01



Protocolo Experimental

Características (Importancia)



		Hiperparametros						
Cantidad Variables	Standarization	Solver	max-iter	random_state	hidden_layer_sizes	alpha	activation	learning_rate_init
18	SI	sgd	2000	1	(10,12,14)	1	relu	0.01
18	SI	sgd	2000	1	(10,12,14,16)	1	relu	0.01
18	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
19	SI	sgd	2000	1	(10,12,14)	1	relu	0.01
19	SI	sgd	2000	1	(10,12,14,16)	1	relu	0.01
19	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01
32	SI	sgd	2000	1	(10,12,14)	1	relu	0.01
32	SI	sgd	2000	1	(10,12,14,16)	1	relu	0.01
32	SI	sgd	2000	1	(10,12,14,16,18)	1	relu	0.01

Secciones Individuales

Resultados

Ejecutamos 12 iteraciones de la metodología CRISP DM. En cada iteración ejecutamos modelos distintos hasta llegar al objetivo planteado de precisión y sensibilidad. Las principales técnicas utilizadas son balanceo de datos, combinaciones de hiperparámetros, estandarización de los datos, análisis PCA.

Iteración 1: ejecución del modelo con características completas sin estandarizar y con datos desbalanceados.

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%

Iteración 4: ejecución del modelo con características completas sin estandarizar y con datos balanceados por sobre muestreo en clase minoritaria.

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

Iteración 9: ejecución del modelo con características completas variando hiperparámetros mediante algoritmo de búsqueda exhaustiva

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

Iteración 12: ejecución del modelo seleccionando con características completas estandarizadas con los mejores hiperparámetros y con datos balanceados mediante algoritmo Smote-Tomek.

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%

Resultados

Con el fin de garantizar la estabilidad del modelo decidimos realizar un análisis final de validación cruzada con dos estrategias: **K iteraciones** y **K iteraciones estratificado**, con los siguientes resultados.

Nota: Es importante aclarar que solo variamos de manera aleatoria el set de datos pero tanto hiperparámetros como características son constantes.

Cross Validation K-Folds

```
[ ]  1 from sklearn.model_selection import cross_val_score
     2 from sklearn.model_selection import KFold
```

```
[ ]  1 kf = KFold(n_splits=3)
     2 scores = cross_val_score(mlp_PRO_FLA, X_trainPF, y_trainPF, cv=kf, scoring="accuracy",n_jobs=-1)
     3 sensibilidades = cross_val_score(mlp_PRO_FLA, X_trainPF, y_trainPF, cv=kf, scoring="recall",n_jobs=-1)
```

```
[ ]  1 print("Metricas scores", scores)
     2 print("Metricas sensibilidades", sensibilidades)
```

```
Metricas scores [0.85386141 0.86167576 0.86024349]
Metricas sensibilidades [0.79596561 0.79314917 0.79787   ]
```

Resultados

Con el fin de garantizar la estabilidad del modelo decidimos realizar un análisis final de validación cruzada con dos estrategias: **K iteraciones** y **K iteraciones estratificado**, con los siguientes resultados.

Nota: Es importante aclarar que solo variamos de manera aleatoria el set de datos pero tanto hiperparámetros como características son constantes.

Cross Validation Stratified K-Fold

```
[ ]  1 from sklearn.model_selection import StratifiedKFold
```

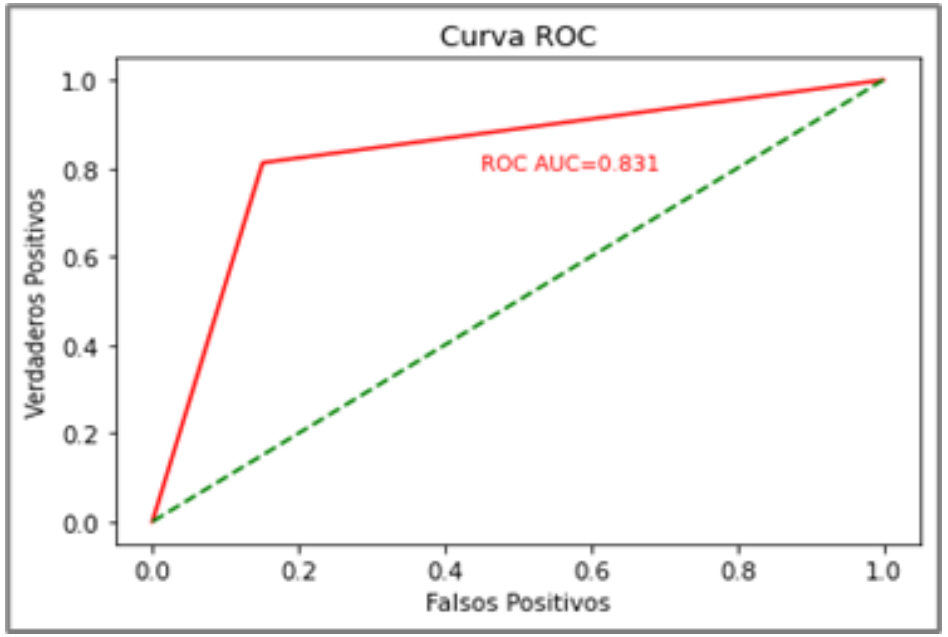
```
 1 skf = StratifiedKFold(n_splits=50, shuffle=True, random_state=1)
 2 lista_accuracy = []
 3 lista_sensibilidad =[]
 4 for train_index, test_index in skf.split(X_trainPF, y_trainPF):
 5     x_train_fold, x_test_fold = X_scaler[train_index], X_scaler[test_index]
 6     y_train_fold, y_test_fold = y_train_smt[train_index], y_train_smt[test_index]
 7     mlp_PRO_FLA.fit(x_train_fold, y_train_fold)
 8     lista_accuracy.append(mlp_PRO_FLA.score(x_test_fold, y_test_fold))
 9     lista_sensibilidad.append(recall_score(y_test_fold,mlp_PRO_FLA.predict(x_test_fold)))
```

```
Lista de socres: [0.8715596330275229, 0.8899082568807339, 0.8743119266055046, 0.8678899082568807, 0
El Maximo accuracy del modelo es: 89.80716253443526 %
El Minimo accuracy del modelo es 84.20569329660239 %
El Promedio accuracy del modelo es: 86.66727323274446 %
La desviacion estandar del accuracy del modelo es: 0.011355373044566392
```

Resultados

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 originales se obtiene la siguiente matriz de confusión y la siguiente curva ROC:

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



Resultados - Topología

F1- Mejores Hiperparámetros

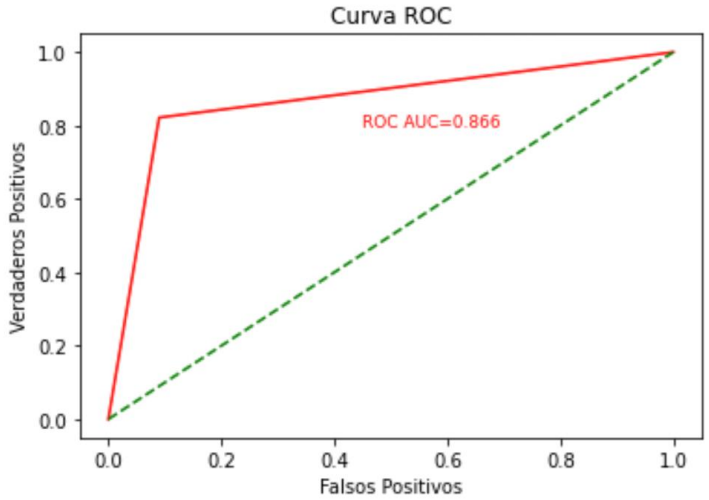
Cantidad de variables	Estandarización	solver	max-iter	Capas(neuronas)	Alpha	Activación	Learning rate	Método balanceo
32	Standart Scaler	sgd	5000	(10,12,14,16,18) 5 capas	1	relu	0,01	smote-Tomek

		Valor	Predicho
	-	no	yes
Valor	no	10621	1049
Verdadero	yes	2081	9589

F2-Matriz de confusión

Salida		
Accuracy train	accuracy score	Sensibilidad
0.8649	0.8658	0.9013

F3 - Scores



F4 - ROC

Conclusiones

- Logramos indicadores de precisión y sensibilidad superiores a los artículos de la bibliografía.
- Las redes neuronales artificiales son capaces de 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 generan sobre ajuste como lo son variables altamente correlacionadas con la variable objetivo o un desbalanceo.
- Es necesario probar diferentes técnicas en cada etapa del desarrollo, a veces los resultados suelen favorecer a una 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 en los estadísticos evaluados 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.

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-over-view (accessed Mar. 20, 2022).
- [6]J. Ariel, C. Ochoa, J. Francisco, and M. Trinidad, “SMOTE-D, UNA VERSIÓN DETERMINISTA DE SMOTE.”

Bibliografía

- [7]“Acerca de las familias de máquinas|Documentación de ComputeEn-gine|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_net-work.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 -Confer-ence Track Proceedings, 2015.
- [10]“sklearn.ensemble.RandomForestClassifier —scikit-learn 1.0.2 documenta-tion.” https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.Ran-domForestClassifier.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 Manage-ment,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 Ad-vanced Technology (IJEAT),2(6).

Bibliografía

- [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
- [17] “CDT, la inversión más rentable en épocas de incertidumbre | Contenido Patrocinado | Portafolio.” https://www.portafolio.co/contenido-patrocinado/cdt-la-inversion-mas-rentable-en-epocas-de-incertidumbre-562982 (accessed Mar. 21, 2022).
- [18] O. Apampa, “Evaluation of Classification and Ensemble Algorithms for Bank Customer O. Apampa Evaluation of Classification and Ensemble Algorithms for Bank Customer Marketing Response Prediction.”

Bibliografía

- [19] E. A. Duran, A. Pamukcu, and H. Bozkurt, “COMPARISON OF DATA MINING TECHNIQUES FOR DIRECT MARKETING CAMPAINGS.”

Redes Neuronales