

Utilización de una red neuronal multicapa para la predicción de operaciones de telemarketing bancario

Introducción

Para el presente trabajo se contó con 35.010 registros de llamados telefónicos realizados entre mayo de 2008 y junio de 2013. Cada registro contiene el resultado de la llamada, variable objetivo (“éxito” o “fracaso”), además de otros 20 atributos inclusive la duración de la llamada. El objetivo del presente trabajo es poder predecir la variable objetivo con un grado de certeza aceptable.

Se analizaron tres posibles casos teniendo en cuenta la presencia de datos desconocidos y el atributo respectivo a la duración de la llamada (DLL). En el primer caso se descartaron los registros en los que hubiere algún dato desconocido además del atributo DLL reduciéndose a 25937 registros. Para el segundo se utilizaron los 25937 registros sin datos desconocidos y se agregó el atributo DLL. En último caso se utilizaron todos los registros inclusive los que contenían datos desconocidos y se eliminó el atributo DLL. En todos los casos la red neuronal (RN) fue construida utilizando la librería Keras con TensorFlow como backend.

Implementación y resultados

A continuación se describe el diseño de la RN y su implementación:

Primer capa: 19 neuronas (caso uno y tres), 20 neuronas (caso dos). Relu como función de activación.

Segunda capa: 9 neuronas, Relu como función de activación.

Tercer capa: 3 neuronas, Relu como función de activación

Capa de salida: 1 neurona, Sigmoidea como función de activación.

En los tres casos se utilizó el error cuadrático medio como función de coste y el algoritmo de optimización Adam en su forma por defecto, que utiliza una tasa de aprendizaje de 0.001. Se dividieron los datos en entrenamiento (80%) y testeo (20%). Durante la fase del entrenamiento se utilizó un set de validación (20% del set de entrenamiento) y se computó el error cuadrático medio del set de validación y entrenamiento por cada época tal como muestran las figuras 1,2 y 3. Se establece un máximo de 700 épocas y se utilizó un frenado anticipado en el caso de que el error cuadrático medio sobre el set de validación no disminuyera en 10 épocas.

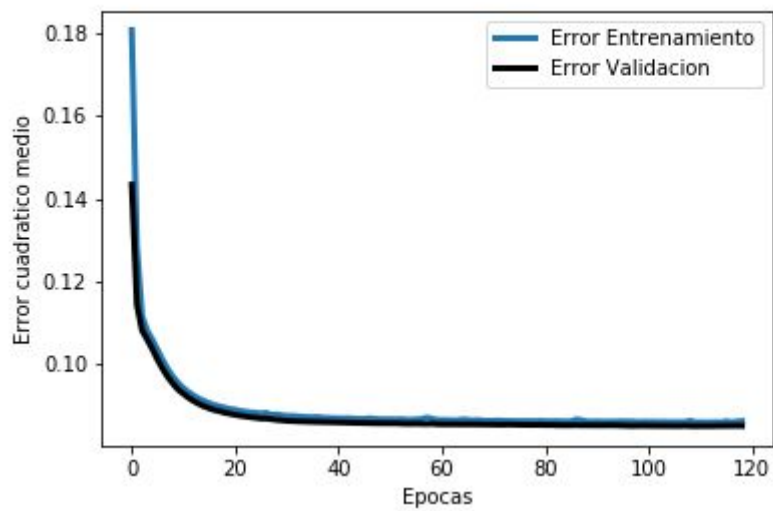


Figura 1: Error cuadrático medio para el set de validación y entrenamiento para el caso 1

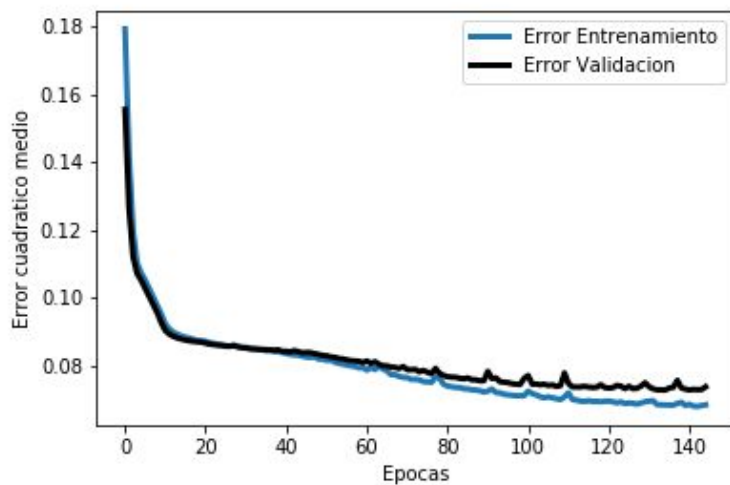


Figura 2: Error cuadrático medio para el set de validación y entrenamiento para el caso 2

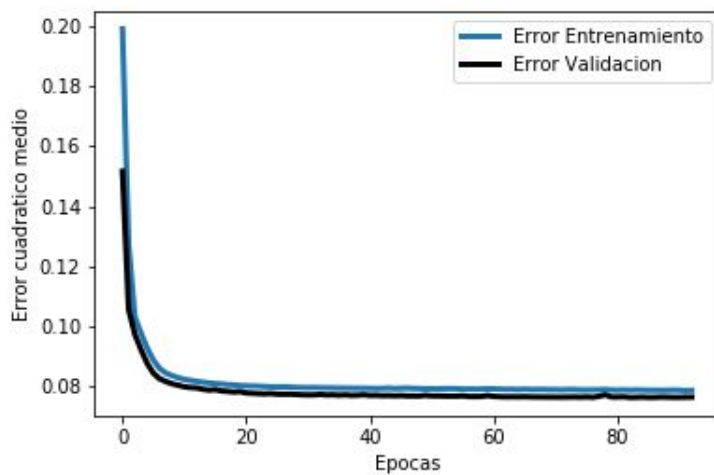


Figura 3: Error cuadrático medio para el set de validación y entrenamiento para el caso 3

En los tres casos se puede observar una rápida disminución del error cuadrático medio para los datos de entrenamiento y validación sin que se pueda observar un sobre ajuste. Se realizaron múltiples ensayos modificando la cantidad de neuronas por capa, la cantidad de capas, la función de coste y el algoritmo de optimización. Se encontraron resultados muy similares en muchos ensayos sin que ninguno mejore los presentados en el presente informe.

Luego de haber entrenado y conseguido la mejor RN posible se realizaron las predicciones sobre los datos de testeo. A continuación en las figuras 4, 5 y 6 se muestran las matrices de confusión para cada caso utilizando un umbral de decisión de 0.5 sobre la variable objetivo. También se detallan las métricas de performance para cada caso.

Matriz de confusión y métricas de performance.

Caso 1

Valores Reales	Si	4437	92
	No	513	146
		Si	No
		Valores Predichos	

Figura 4: Matriz de confusión para el caso 1.

Métricas de performance

Accuracy, qué tan seguido clasifica bien?

$$(VP+VN)/TOTAL=(4437+146)/5188=0.88$$

Tasa de error, que tan seguido clasifica mal?

$$1-Accuracy=1-0.88=0.12$$

Tasa de Verdaderos Positivos o Sensibilidad, si el valor real es positivo que tan seguido lo predice?

$$VP/(VP+FN)=4437/(4437+92)=0.98$$

Tasa de Falsos positivos, si el valor real es negativo que tan seguido predice positivo?

$$FP/(FP+VN)=513/(513+146)=0.78$$

Tasa de verdadero negativo o Especificidad, si el valor real es negativo que tan seguido lo predice?

$$VN/(FP+VN)=146/(513+146)=0.22$$

Caso 2

Valores Reales	Si	4287	242
	No	290	369
		Si	No
		Valores Predichos	

Figura 5: Matriz de confusión para el caso 2.

Métricas de performance

Accuracy

$$(VP+VN)/TOTAL=(4287+369)/5188=0.90$$

Tasa de error

$$1-Accuracy=1-0.90=0.10$$

Tasa de Verdaderos Positivos o Sensibilidad

$$VP/(VP+FN)=4287/(4287+242)=0.95$$

Tasa de Falsos positivos

$$FP/(FP+VN)=290/(290+369)=0.44$$

Tasa de verdadero negativo o Especificidad

$$VN/(FP+VN)=369/(290+369)=0.56$$

Caso 3

Valores Reales	Si	6110	91
	No	621	180
		Si	No
		Valores Predichos	

Figura 6: Matriz de confusión para el caso 3.

Métricas de performance

Accuracy

$$(VP+VN)/TOTAL=(6110+180)/7002=0.90$$

Tasa de error

$$1-Accuracy=1-0.90=0.10$$

Tasa de Verdaderos Positivos o Sensibilidad

$$VP/(VP+FN)=6110/(6110+91)=0.98$$

Tasa de Falsos positivos

$$FP/(FP+VN)=621/(621+180)=0.77$$

Tasa de verdadero negativo o Especificidad

$$VN/(FP+VN)=180/(180+621)=0.22$$

Como se mencionó previamente la matriz de confusión y las métricas mostradas fueron calculadas con un umbral de decisión de 0.5. Una manera de resumir la performance de un clasificador binario sobre todos los posibles umbrales es mediante una curva ROC (acrónimo de Característica Operativa del Receptor). La curva ROC muestra la representación de la razón o ratio de verdaderos positivos frente al ratio de falsos positivos para todos los posibles umbrales de decisión. De esta manera se puede buscar la relación deseada entre la tasa de verdaderos positivos y la tasa de falsos positivos para luego encontrar el umbral que satisface dicha condición. Además el área debajo de la curva (AUC, acrónimo de siglas en inglés) indica cuanto el modelo es capaz de distinguir entre dos clases. Un AUC de 1 indica que el modelo es perfectamente capaz de distinguir dos clases mientras que cuanto más se acerque a 0.5 su capacidad de distinción será azarosa. En las figuras 7, 8 y 9 se muestran las curvas ROC para los tres casos. Se encuentra una gran similitud entre las curvas ROC y las AUC del caso 1 y 3. Mientras que en el caso 2 se observa una notable mejora del modelo, al utilizar el atributo DLL.

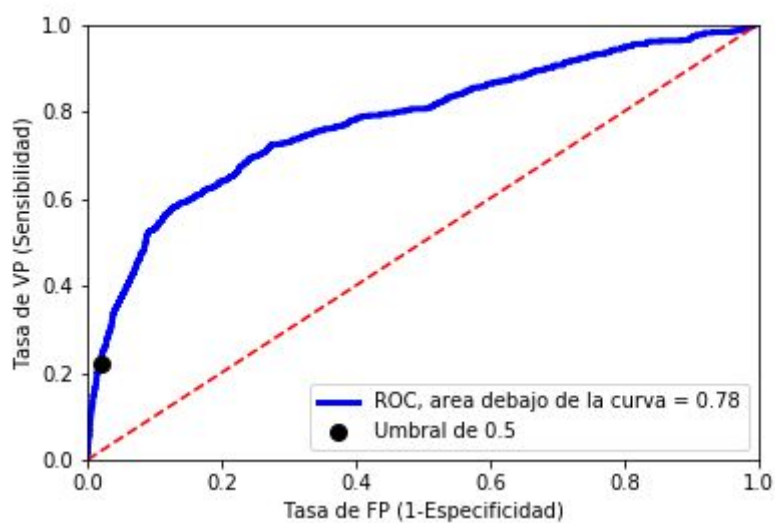


Figura 7: Para el caso 1 curva ROC en azul, el punto negro indica un umbral de 0.5 y $AUC=0.78$. La línea punteada indica la curva ROC de un modelo azaroso con $AUC=0.5$

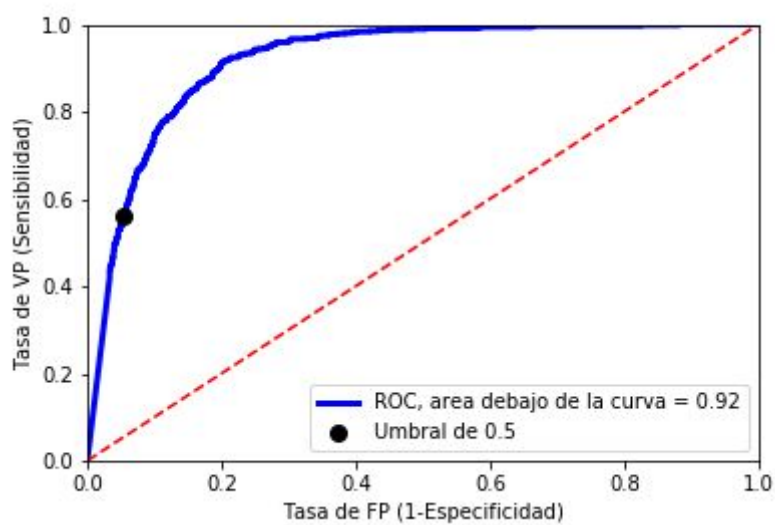


Figura 8: Para el caso 2 curva ROC en azul, el punto negro indica un umbral de 0.5 y $AUC=0.92$. La línea punteada indica la curva ROC de un modelo azaroso con $AUC=0.5$

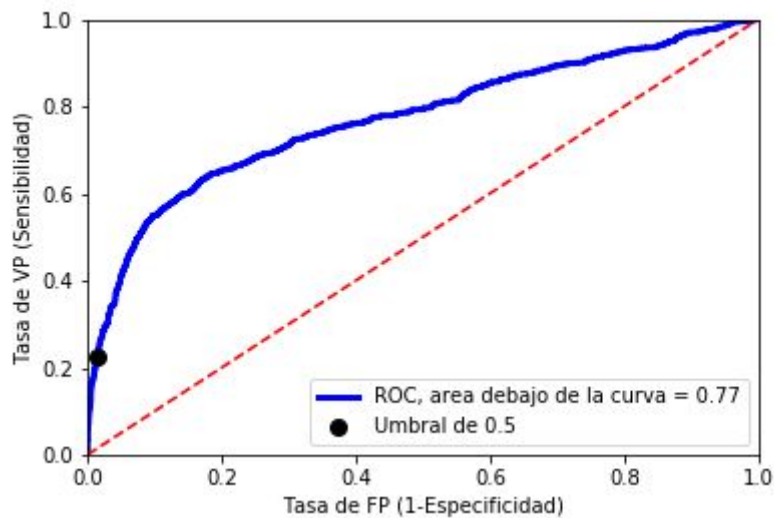


Figura 9: Para el caso 3 curva ROC en azul, el punto negro indica un umbral de 0.5 y AUC=0.77. La línea punteada indica la curva ROC de un modelo azaroso con AUC=0.5

Conclusiones

Se puede concluir que la presencia de datos desconocidos no condiciona a la RN, obteniéndose resultados de gran similaridad entre los casos 1 y 3. Para los casos 1 y 3 la mejora esperada al utilizar una RN para identificar operación telefónica exitosas es del 27 % sobre la ausencia de modelo predictivo. Si bien a la hora de realizar una campaña de marketing no se cuenta con la información del atributo DLL, incluirlo en la RN genera una mejora sustancial. En trabajos previos (Moro et. al 2014.) se utilizó una RN sobre un set de datos de gran similaridad en proporción y atributos al utilizado en el presente trabajo y se llegó a resultados iguales para el caso 3. Indicando que los resultados aquí presentados se encuentran en línea con los ya obtenidos en trabajos previos.