Airlines Passenger Satisfaction Classifier with Neural Networks

German Salinas[1], Renata Victoria Catelli[2] and Rocio Belén Tantos[3]

Universidad Tecnológica Nacional, Facultad Regional La Plata. La Plata, Buenos Aires. Argentina

[1] gsalinas@alu.frlp.utn.edu.ar
[2] renatacatelli@alu.frlp.utn.edu.ar
[3] rtantos@alu.frlp.utn.edu.ar

Abstract. Passenger satisfaction is a key indicator of service quality in the airline industry and plays a crucial role in customer retention and competitiveness. This paper presents a neural network-based classifier designed to predict passenger satisfaction using airline operational data and service quality attributes. The proposed approach follows a supervised learning paradigm and includes several data preprocessing steps like data cleaning, normalization, and categorical feature encoding. The model is trained on a labeled passenger satisfaction dataset and evaluated using standard classification metrics to assess its predictive performance. The results show that neural networks are effective in capturing complex, non-linear relationships between service-related features and overall passenger satisfaction. Additionally, the analysis of model output provides insights into the service factors that most strongly influence passenger perceptions. These findings demonstrate the potential of neural networks models as decision-support tools for airlines, enabling data-driven identification of areas for service improvement. By leveraging passenger feedback and operational data, airlines can better understand customer needs and enhance their overall travel experience.

Keywords: Artificial Neural Network, Backpropagation, Binary Classification, Passenger satisfaction, Supervised learning.

Clasificación de la Satisfacción de Pasajeros Aéreos mediante Redes Neuronales Artificiales

Resumen. La satisfacción de los pasajeros es un indicador clave de la calidad del servicio en el sector aéreo y desempeña un papel fundamental en la retención de clientes y la competitividad. Este artículo presenta un clasificador basado en redes neuronales artificiales para predecir la satisfacción de los pasajeros a partir de datos operativos y atributos de calidad del servicio. El modelo, entrenado bajo un paradigma de aprendizaje supervisado con preprocesamiento, normalización y balanceo de clases, se evalúa mediante métricas estándar de clasificación. Los resultados muestran que las redes neuronales capturan eficazmente las relaciones

no lineales entre las características del servicio y la satisfacción general, constituyendo una herramienta de apoyo a la toma de decisiones que permite a las aerolíneas identificar, basándose en datos, las áreas de mejora del servicio.

Palabras Claves: Red Neuronal Artificial, Backpropagation, Clasificación Binaria, Satisfacción de Pasajeros, Aprendizaje Supervisado.

1. Introducción

En un mercado aéreo cada vez más competitivo, la experiencia del pasajero se ha consolidado como un factor clave para la fidelización y la reputación de las aerolíneas, y la calidad del servicio ha sido identificada como un determinante central de las intenciones de comportamiento de los pasajeros (Park, Robertson y Wu, 2004). Comprender los elementos que influyen en la satisfacción de los clientes trasciende el plano de la percepción subjetiva y se constituye como un problema de análisis de datos a gran escala, en el que la inteligencia artificial permite transformar la opinión de los pasajeros en información accionable, capaz de respaldar decisiones estratégicas y mejoras concretas en la calidad del servicio.

En el presente artículo se desarrolla un sistema basado en Redes Neuronales Artificiales (RNA) para clasificar el nivel de satisfacción de pasajeros aéreos a partir de variables operativas, demográficas y vinculadas al servicio brindado. Se implementa un modelo de aprendizaje supervisado entrenado mediante *backpropagation*, orientado a una clasificación binaria que distingue pasajeros satisfechos e insatisfechos, sobre un conjunto de datos público de la plataforma *Kaggle* al que se aplican técnicas de preprocesamiento, balanceo de clases y normalización de variables.

El objetivo del trabajo consiste en comparar distintas configuraciones de RNA aplicadas al problema, evaluando el impacto de estrategias de regularización, balanceo de clases, selección de variables y funciones de activación sobre el desempeño del modelo.

El artículo se organiza de la siguiente manera: la Sección 2 presenta el marco teórico y los trabajos relacionados; la Sección 3 describe la metodología empleada, incluyendo el modelo propuesto, su arquitectura, el conjunto de datos y los criterios de éxito; la Sección 4 expone los resultados obtenidos; la Sección 5 discute dichos resultados; y finalmente, la Sección 6 presenta las conclusiones y líneas futuras de investigación.

2. Marco Teórico

El trabajo se enmarca en el campo del aprendizaje automático supervisado, en el que un modelo aprende patrones a partir de ejemplos etiquetados (Goodfellow, Bengio y Courville, 2016). Se utiliza una RNA de tipo *feedforward*, donde la información fluye en una única dirección desde la capa de entrada hacia la de salida, entrenada mediante el algoritmo de *backpropagation* (Rumelhart, Hinton y Williams, 1986). Este tipo de arquitectura es capaz de capturar relaciones no lineales y patrones complejos presentes

en los datos (LeCun, Bengio y Hinton, 2015). Al tratarse de una clasificación binaria, la capa de salida emplea la función sigmoide, cuya salida se interpreta como la probabilidad de pertenencia a una de las dos clases, mientras que en las capas ocultas se utilizan las funciones *ReLU* (Nair y Hinton, 2010) y *tanh*.

El problema de la predicción de la satisfacción de pasajeros aéreos cuenta con antecedentes en la literatura. Noviantoro y Huang (2022) compararon distintos algoritmos de minería de datos sobre el mismo conjunto de datos utilizado en este trabajo, obteniendo los mejores resultados con métodos de ensamble como *random forest*. Por su parte, Hayadi et al. (2021) evaluaron diversos clasificadores supervisados para esta misma tarea. Estos antecedentes confirman la viabilidad del enfoque basado en aprendizaje automático y proporcionan un punto de comparación para los resultados obtenidos.

Dos desafíos metodológicos guían el diseño experimental. El primero es el desbalanceo de clases del *dataset*, que posee más registros de pasajeros insatisfechos que satisfechos; para abordarlo se aplican el sobremuestreo aleatorio (*oversampling*) de la clase minoritaria y la ponderación de clases durante el entrenamiento (He y Garcia, 2009). El segundo es el riesgo de sobreajuste (*overfitting*), mitigado mediante *dropout*, que desactiva aleatoriamente neuronas en cada iteración (Srivastava et al., 2014), y *early stopping*, que detiene el entrenamiento cuando el rendimiento sobre el conjunto de validación deja de mejorar (Prechelt, 2012).

El preprocesamiento incluye la normalización de las variables numéricas para que tengan media cero y desviación estándar uno, evitando que aquellas de mayor escala dominen el aprendizaje. El desempeño se evalúa con métricas estándar de clasificación —matriz de confusión, precisión, *recall* y *F1-score*—, complementadas con el análisis de las curvas de pérdida y exactitud durante el entrenamiento para detectar síntomas de subajuste o sobreajuste (Goodfellow et al., 2016).

3. Elementos de Trabajo y Metodología

3.1 Modelo de RNA utilizada con la justificación de su elección.

Para resolver el problema de clasificación binaria del nivel de satisfacción de pasajeros aéreos se utiliza una RNA entrenada mediante *backpropagation* (Rumelhart et al., 1986). En cada ciclo de entrenamiento, la red propaga los datos hacia adelante, calcula el error entre la salida generada y la esperada, y lo propaga hacia atrás para ajustar los pesos de conexión mediante descenso del gradiente. Este mecanismo permite aprender representaciones intermedias y generalizar patrones aun frente a datos no observados previamente (LeCun et al., 2015).

Existen métodos alternativos igualmente aplicables a la clasificación binaria sobre datos tabulares, como la regresión logística, los árboles de decisión o *random forest* (Breiman, 2001), que en estudios previos sobre este mismo problema alcanzaron resultados competitivos (Noviantoro y Huang, 2022; Hayadi et al., 2021). Frente a ellos, la RNA ofrece mayor flexibilidad para modelar interacciones no lineales entre los atributos del servicio y permite estudiar de forma controlada el efecto de distintas decisiones de diseño —regularización, balanceo de clases, selección de variables y

funciones de activación— sobre una misma arquitectura, lo que constituye el objetivo central de este trabajo. Como contrapartida, presenta menor interpretabilidad y mayor costo de ajuste de hiperparámetros, por lo que la comparación sistemática con dichos métodos se plantea como línea de trabajo futuro.

3.2 Arquitectura y topología final de la RNA

Modelo 1 – Oversampling con early stopping. El modelo seleccionado utiliza la API *Sequential* de *Keras* para construir la RNA como una pila lineal de capas, estructura que se adapta al flujo unidireccional de las redes *feedforward* clásicas. A continuación, se detallan los componentes de la arquitectura empleada:

La arquitectura se compone de una capa de entrada densa de 23 neuronas (una por variable predictora del conjunto de entrenamiento), dos capas ocultas de 32 neuronas totalmente conectadas con activación *ReLU* —cada una seguida de una capa de *dropout* del 30% para reducir el riesgo de sobreajuste (Srivastava et al., 2014)— y una capa de salida de una única neurona con activación sigmoide, que retorna la probabilidad de que el pasajero esté satisfecho.

Compilación del modelo. Se emplea la función de pérdida *binary crossentropy* (entropía cruzada binaria), adecuada para la clasificación binaria; el optimizador *Adam*, un algoritmo adaptativo que ajusta la tasa de aprendizaje de cada parámetro (Kingma y Ba, 2015); y la exactitud (*accuracy*) como métrica de monitoreo durante el entrenamiento y la validación.

Hiperparámetros de entrenamiento. Se establece un límite inicial de 100 épocas con lotes (*batch size*) de 64 muestras, equilibrando estabilidad del gradiente y eficiencia de cómputo. Se aplica *early stopping* con una paciencia de 10 épocas, interrumpiendo el entrenamiento cuando el rendimiento sobre el conjunto de validación deja de mejorar (Prechelt, 2012).

3.3 Patrones utilizados para el entrenamiento y la validación de la RNA

Se utiliza el *dataset* público "Airline Passenger Satisfaction" (Mahal, 2020) de la plataforma *Kaggle*, compuesto por 129.880 registros y 25 columnas, de las cuales 22 son variables predictoras y una corresponde a la variable objetivo (*satisfaction*), que indica si un pasajero está satisfecho con su experiencia de vuelo. Las variables predictoras se agrupan en características del servicio a bordo (15 variables, como comida, entretenimiento, confort del asiento, limpieza y embarque), demográficas (5 variables, como edad, género, clase y tipo de cliente) y operativas (2 variables: distancia del vuelo y minutos de demora); se descartan dos columnas sin valor predictivo (*Unnamed* e *Id*).

Originalmente, el *dataset* está dividido en archivos de entrenamiento y prueba. Sin embargo, se opta por concatenarlos y realizar una nueva partición estratificada del conjunto unificado: 75% para entrenamiento y 25% para validación, manteniendo la distribución proporcional de las clases.

La variable objetivo original contempla las categorías "Satisfied" y "Neutral or Dissatisfied"; esta última se renombra como "Dissatisfied" para abordar la tarea de

clasificación binaria y priorizar la detección temprana de la insatisfacción de los pasajeros.

Si bien el *dataset* utilizado es ampliamente empleado en tareas de aprendizaje automático, su procedencia y proceso de recolección no se encuentran completamente documentados. Además, presenta un nivel de preprocesamiento previo y una estructura organizada que puede diferir de escenarios reales de operación aérea. Por lo tanto, los resultados obtenidos deben interpretarse dentro de un entorno controlado de experimentación. La descripción detallada del dataset se encuentra referenciada y disponible en la documentación oficial de la plataforma *Kaggle*.

Preprocesamiento. Las variables numéricas se escalan mediante *StandardScaler* para que tengan media cero y desviación estándar uno, evitando que aquellas de mayor rango dominen el proceso de aprendizaje. Frente al desbalance de clases —cerca de 73.000 pasajeros insatisfechos y 56.000 satisfechos—, se aplica sobremuestreo aleatorio de la clase minoritaria, opción priorizada por sobre el submuestreo, ya que eliminar registros de la clase mayoritaria implicaría pérdida de información potencialmente relevante.

3.4 Herramientas utilizadas

El desarrollo se realiza en *Python* sobre *Google Colab*. La manipulación y el análisis exploratorio de los datos se llevan a cabo con *Pandas* y *NumPy*, junto con *Matplotlib* y *Seaborn* para la visualización; el modelado y la evaluación emplean *Scikit-learn* (Pedregosa et al., 2011) e *Imbalanced-learn* para el tratamiento del desbalanceo de clases, y la RNA se construye y entrena con *TensorFlow* (Abadi et al., 2016) y su API *Keras*.

3.5 Características de los distintos prototipos

Durante el desarrollo del artículo se evalúan diferentes variantes del modelo principal con el fin de analizar su comportamiento en distintas condiciones. Todos los modelos comparten la misma arquitectura base, a excepción de uno.

Modelo 2 – Oversampling sin early stopping. Esta variante replica la arquitectura del modelo principal, pero sin el mecanismo de *early stopping*, con el objetivo de evaluar cómo influye en el proceso de entrenamiento la ausencia de esta técnica de regularización.

Modelo 3 – Selección de variables con mayor correlación positiva. En esta variante, la diferencia central radica en la reducción del conjunto de variables predictoras, seleccionando únicamente las siete características con mayor correlación positiva respecto a la variable objetivo.

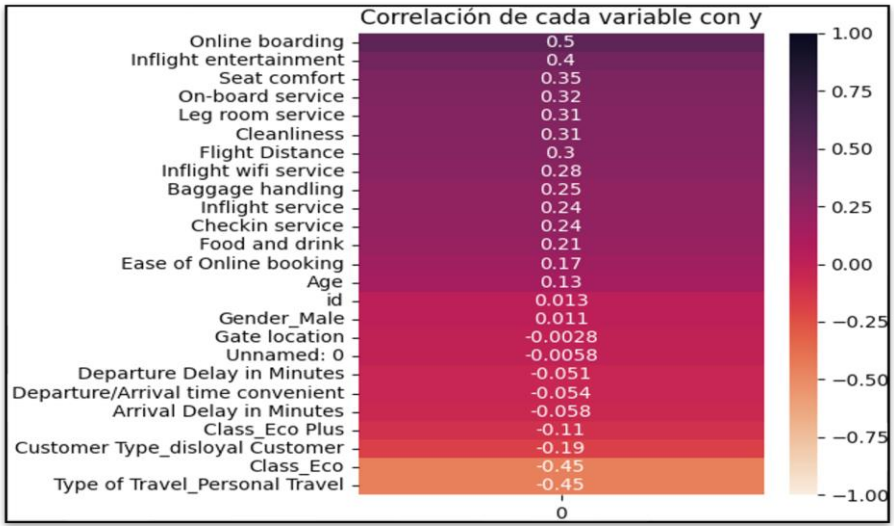


Fig. 1. Matriz de correlación de variables respecto a la satisfacción

La selección de atributos se basa en un análisis exploratorio de correlación (Fig. 1), tomando aquellos con una correlación positiva superior a 0,30 respecto de la variable objetivo. Esta estrategia busca reducir la dimensionalidad y el tiempo de entrenamiento, y evaluar si las variables más influyentes son suficientes para lograr un desempeño similar al del modelo principal.

Modelo 4 – Selección de variables con correlación positiva y negativa. Esta variante mantiene la estructura del Modelo 3, pero adiciona variables con correlación negativa respecto de la satisfacción del pasajero hasta -0.30 (*Type of Travel_Personal Travel y Class_Eco*). El objetivo de este experimento es evaluar si estas variables con ambos tipos de correlación son capaces de aportar información predictiva útil, quitando variables no relevantes para el caso de estudio.

Modelo 5 – Activación *tanh* y balanceo con *class_weight*. Implementa tres capas ocultas de 64, 32 y 16 neuronas con activación *tanh* —centrada en cero, con rango [-1, 1]— y sigmoide en la salida. Además, reemplaza el sobremuestreo por el parámetro *class_weight='balanced'* de *Scikit-learn*, que penaliza en mayor medida los errores cometidos sobre la clase minoritaria, buscando un mejor equilibrio entre precisión y *recall* sin los posibles inconvenientes del sobremuestreo.

3.6 Criterios de éxito

Previo a la experimentación se definen los criterios de éxito del modelo: (i) una exactitud (*accuracy*) general mínima del 80%, complementada con otras métricas debido al desbalance de clases; (ii) una matriz de confusión con bajos falsos positivos y negativos, sin sesgos marcados hacia una clase, priorizando la correcta detección de

pasajeros insatisfechos; (iii) un equilibrio entre precisión y *recall*, con énfasis en la clase "insatisfecho", dado que su correcta clasificación permite a la aerolínea actuar sobre posibles mejoras del servicio; y (iv) curvas de pérdida y exactitud coherentes entre entrenamiento y validación, sin divergencias que indiquen sobreajuste o subajuste.

4. Resultados

4.1 Evolución del entrenamiento

Durante el entrenamiento del Modelo 1 se aplica *early stopping* con los parámetros *monitor='val_loss'*, *patience=10* y *restore_best_weights=True*, deteniendo el proceso al estabilizarse la métrica de validación y conservando los pesos correspondientes al mejor desempeño observado.

El entrenamiento, planificado para un máximo de 100 épocas, se detiene automáticamente en la época 79, cuando el modelo muestra señales de convergencia y estabilidad. El valor final de *loss* es de 0,1264 en entrenamiento y 0,1177 en validación, lo que evidencia un equilibrio entre ambos conjuntos y sugiere ausencia de sobreajuste (Tabla 1).

Tabla 1. Evolución del modelo durante su ejecución

Época	Accuracy (entrenamiento)	Accuracy (validación)	Loss (entrenamiento)	Loss (validación)
1	73.29%	88.35%	0.5438	0.2780
35	94.45%	95.32%	0.1369	0.1118
77	94.71%	95.56%	0.1271	0.1056
78	94.62%	95.21%	0.1257	0.1079
79	94.73%	94.76%	0.1264	0.1177

4.2 Métricas de rendimiento global

Tras el entrenamiento, el modelo es evaluado sobre el conjunto de prueba, obteniendo un *Accuracy* del 95%. Este valor demuestra que la RNA es altamente capaz de clasificar correctamente los pasajeros satisfechos e insatisfechos. Además, el equilibrio entre precisión y *recall* para ambas clases confirma su utilidad en contextos donde es prioritario identificar experiencias negativas sin incurrir en falsos positivos excesivos.

4.3 Resultados obtenidos

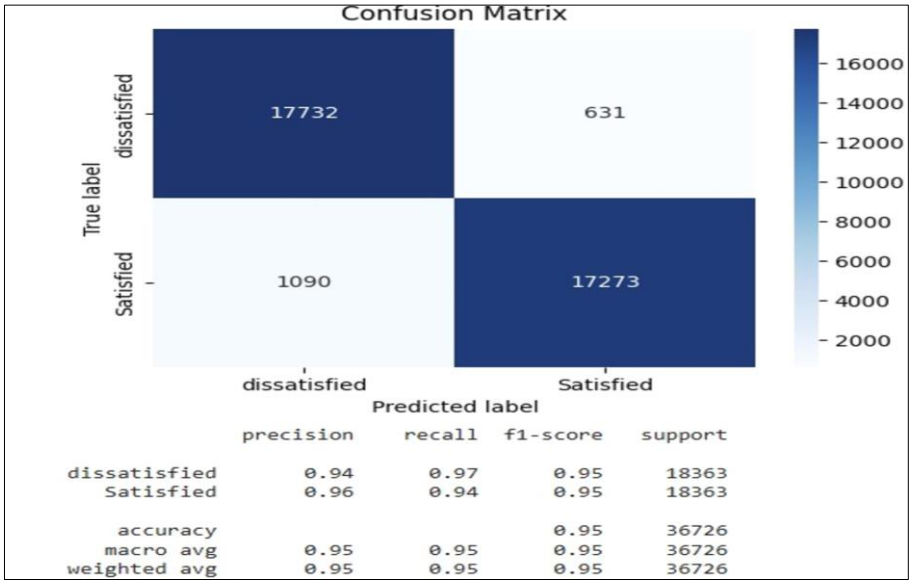
Con los resultados del modelo, la calidad de estos se refleja en las métricas específicas para cada clase. En el caso de los pasajeros insatisfechos, el eje principal del análisis alcanza una alta precisión y *recall*, tal como se muestra en la Fig. 2. Estos

valores indican que el sistema identifica con muy bajo margen de error a pasajeros con experiencias negativas, logrando su detección anticipada y eventual intervención.

	precision	recall	f1-score	support
dissatisfied	0.93	0.97	0.95	18363
Satisfied	0.97	0.93	0.95	18363
accuracy			0.95	36726
macro avg	0.95	0.95	0.95	36726
weighted avg	0.95	0.95	0.95	36726

Fig. 2. Métricas del modelo.

Del total de muestras del conjunto de prueba, 35.005 predicciones son correctas y 1.721 incorrectas. La matriz de confusión (Fig. 3) muestra errores controlados, con un leve desbalance entre falsos positivos (631 casos) y falsos negativos (1.090 casos) que no compromete la validez del modelo. En este sentido, se prioriza el *recall* por sobre la precisión en la clase minoritaria, alineado con el criterio de minimizar el riesgo de no detectar a un pasajero disconforme.



	precision	recall	f1-score	support
dissatisfied	0.94	0.97	0.95	18363
Satisfied	0.96	0.94	0.95	18363
accuracy			0.95	36726
macro avg	0.95	0.95	0.95	36726
weighted avg	0.95	0.95	0.95	36726

Fig. 3. Matriz de confusión del modelo sobre el conjunto de prueba.

Las curvas de entrenamiento evidencian un descenso gradual de la función de pérdida y una estabilización coherente de la exactitud en las fases de entrenamiento y validación. La proximidad entre curvas denota un aprendizaje efectivo, sin señales de sobreajuste.

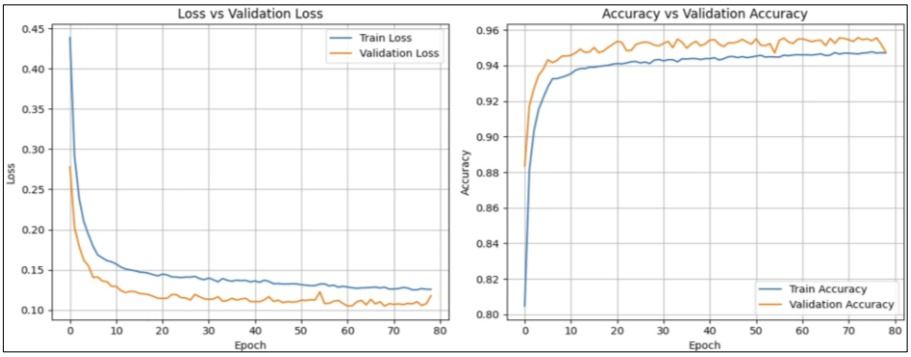


Fig. 4. Curvas de evolución de la pérdida y la exactitud durante el entrenamiento del modelo

5. Discusión

A partir de los criterios de éxito definidos en la Sección 3, en esta sección se analiza el desempeño del modelo principal y de sus variantes experimentales, junto con su aplicabilidad práctica.

5.1. Desempeño del Modelo 1 en la clasificación de satisfacción

Los resultados confirman que el Modelo 1 – *Oversampling* con *early stopping* cumple con los criterios de éxito definidos: la RNA clasifica eficazmente la satisfacción a partir de datos operativos, demográficos y de servicio, con un 95% de *accuracy* y curvas de *loss* y *accuracy* estables que descartan tanto el sobreajuste como el subajuste. En conjunto, el modelo se demuestra robusto y aplicable, constituyendo una base sólida para futuras mejoras.

5.2. Comparación de desempeño entre modelos

En la fase experimental se analizan distintas configuraciones de la arquitectura seleccionada con el objetivo de comparar su desempeño. Si bien todas las variantes presentan resultados aceptables, el modelo elegido como óptimo exhibe, en términos generales, el mejor equilibrio entre rendimiento y estabilidad.

Modelo 2 – *Oversampling* sin *early stopping*. Presenta resultados similares al modelo principal, aunque hacia las últimas épocas se observa una disociación entre entrenamiento y validación (Fig. 5), lo que evidencia la importancia del *early stopping* para lograr entrenamientos más estables y controlados.

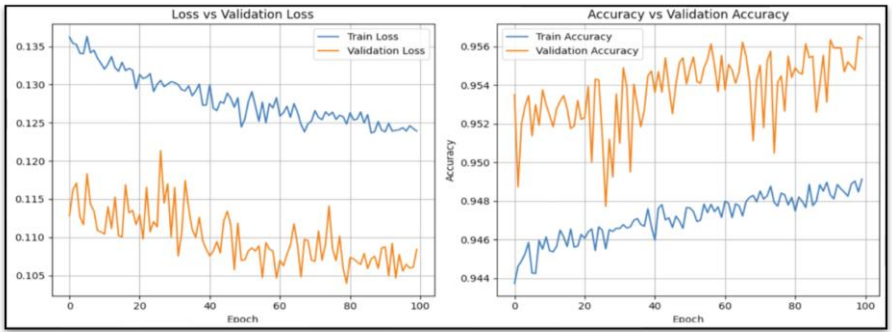


Fig. 5. Evolución de la pérdida y la exactitud en el entrenamiento y la validación.

Modelo 3 – Selección de variables con mayor correlación positiva. Presenta una degradación del rendimiento respecto de la configuración óptima, con una exactitud global del 86%, 2.152 falsos positivos y 1.909 falsos negativos (*F1-score* de 0,86 en ambas clases). Esto indica que las variables seleccionadas, aun con alta correlación individual, no son suficientes para capturar la complejidad total del problema.

Modelo 4 – Selección de variables con correlación positiva y negativa. Mejora el desempeño respecto del Modelo 3, con una exactitud del 89% (*F1-score* de 0,90 para *dissatisfied* y 0,89 para *satisfied*), aunque continúa levemente por debajo del modelo principal. Los resultados sugieren que algunas variables eliminadas guardan relaciones ocultas con las seleccionadas, lo que justifica un análisis más profundo en futuras iteraciones.

Modelo 5 – Activación *tanh* y balanceo con *class_weight*. Alcanza un desempeño aceptable, aunque inferior al principal, con una exactitud del 87% y una distribución de errores equilibrada entre clases (*F1-score* de 0,88 para *dissatisfied* y 0,85 para *satisfied*). A nivel computacional, la red más profunda con *tanh* incrementa el costo de entrenamiento y la sensibilidad a los hiperparámetros, resultando menos eficiente cuando se busca alta precisión con bajo costo de ajuste.

6. Conclusión

En este artículo se diseña un sistema basado en RNA que alcanza un alto rendimiento, con métricas equilibradas entre clases que permiten identificar con fiabilidad tanto a los pasajeros satisfechos como a aquellos con experiencias negativas, resultando útil en entornos de decisión orientados a la mejora continua del servicio.

Es importante mencionar que el *dataset* extraído de *Kaggle* presenta cierta preparación previa, con una estructura clara y sin inconsistencias, lo que facilita el entrenamiento del modelo y contribuye a los buenos resultados obtenidos. Esto permite trabajar en un entorno controlado, ideal para validar el enfoque y sentar las bases para

futuras pruebas en contextos más complejos o menos estructurados.

Como cierre, se demuestra cómo la aplicación de redes neuronales artificiales al análisis de satisfacción de pasajeros constituye una solución viable y escalable para la industria aérea, capaz de transformar grandes volúmenes de datos en información accionable para la toma de decisiones estratégicas. Como líneas futuras se plantean la comparación sistemática con métodos alternativos de clasificación —regresión logística, árboles de decisión y *random forest*—, la validación del enfoque con datos operativos reales de aerolíneas y la optimización del equilibrio entre falsos positivos y falsos negativos.

Referencias

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., … Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. En *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (pp. 265–283). USENIX Association.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Catelli, R., Salinas, G., & Tantos, R. (2025). *Clasificación de la satisfacción de pasajeros aéreos mediante redes neuronales artificiales* [Cuaderno de Google Colab]. Google Colab. https://colab.research.google.com/drive/1TeHInLhwnk9QrkVjQ6lf2aVVEtafpxMq

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hayadi, B. H., Kim, J.-M., & Hulliyah, K. (2021). Predicting airline passenger satisfaction with classification algorithms. *International Journal of Informatics and Information Systems*, 4(1), 82–94.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. En *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Mahal, T. J. (2020). *Airline passenger satisfaction* [Conjunto de datos]. Kaggle. https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. En *Proceedings of the 27th International Conference on Machine Learning (ICML)* (pp. 807–814).

Noviantoro, T., & Huang, J.-P. (2022). Investigating airline passenger satisfaction: Data mining method. Research in Transportation Business & Management, 43, 100726. https://doi.org/10.1016/j.rtbm.2021.100726

Park, J.-W., Robertson, R., & Wu, C.-L. (2004). The effect of airline service quality on passengers' behavioural intentions: A Korean case study. *Journal of Air Transport Management*, 10(6), 435–439. https://doi.org/10.1016/j.jairtraman.2004.06.001

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Prechelt, L. (2012). Early stopping—But when? En G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (2.ª ed., pp. 53–67). Springer. https://doi.org/10.1007/978-3-642-35289-8_5

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. https://doi.org/10.1038/323533a0

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958.