Instituto Tecnológico y de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias

Campus Monterrey

TC3006C.102: Inteligencia artificial avanzada para la ciencia de datos I

Entregable 2: Reto Selección, configuración y entrenamiento del modelo

Equipo 3:

Fernando Bustos Monsiváis - A00829931

Ramón Yuri Danzos García - A00227838

Axel Amós Hernández Cárdenas - A00829837

Josué Emmanuel Flores Mendoza - A00833132

Jesus Daniel Martínez García - A00833591

Profesores:

Dr. Alfredo Esquivel Jaramillo

Dr. Hugo Terashima Marín

Dr. Julio Antonio Juárez Jiménez

Sábado 31 de agosto de 2024

Abstract

Este documento presenta un análisis exhaustivo sobre la predicción de la supervivencia de los pasajeros del Titanic utilizando diversos modelos de machine learning. El estudio abarca desde el preprocesamiento de datos, pasando por la implementación y configuración de varios algoritmos, hasta la evaluación y comparación de sus rendimientos. Los modelos analizados incluyen Regresión Logística, Random Forest, K-Nearest Neighbors (KNN) y Redes Neuronales. Finalmente, se selecciona el modelo KNN como el más adecuado, basándose en su capacidad para capturar patrones de manera efectiva, ofrecer resultados consistentes y equilibrar la precisión con la simplicidad del modelo. El informe concluye con una discusión sobre las implicaciones de los resultados y sugerencias para futuras mejoras.

Índice

1. Introducción

El hundimiento del Titanic se erige como una de las catástrofes marítimas más notorias en la historia. El 15 de abril de 1912, durante su viaje inaugural, el Titanic, considerado por muchos como insumergible, se hundió tras colisionar con un iceberg. Lamentablemente, de los 2,224 pasajeros a bordo, solo 772 lograron sobrevivir, una tragedia en gran parte atribuible a la insuficiencia de botes salvavidas en la embarcación (Cukierski, 2012).

Aunque la suerte desempeña un papel inevitable en situaciones de vida o muerte, en el caso del Titanic, parece que ciertos grupos de personas tenían mayores probabilidades de supervivencia que otros. Por tanto, resulta de gran interés analizar y determinar, a partir de los datos de los pasajeros, qué tipo de individuos presentaban mayores oportunidades de sobrevivir.

2. Planteamiento del problema

El objetivo de este informe es abordar el proceso de preprocesamiento de datos de manera manual, siguiendo el esquema de *Extract, Transform & Load (ETL)*, con el propósito de identificar los factores clave que podrían influir en la supervivencia de los pasajeros. A continuación, se describe la metodología empleada:

1. En primer lugar, se realizará un análisis exhaustivo del conjunto de datos proporcionado, desglosando sus componentes, incluyendo la variable objetivo (*label*), los tipos de datos y la presencia de valores faltantes. Esto permitirá realizar una comparación con el conjunto de datos original una vez completada la limpieza.
2. Posteriormente, se aplicarán métodos estadísticos y gráficos para determinar y justificar la inclusión o exclusión de ciertos valores. Asimismo, se explicará cómo se gestionarán las variables categóricas, como el sexo, el puerto de embarque, entre otras.
3. Finalmente, se llevará a cabo una comparación entre los datos resultantes y los iniciales, con el fin de evaluar la mejora en la calidad de la información que se utilizará para entrenar el modelo.

Al finalizar estos pasos, se espera que los datos obtenidos sean suficientemente robustos, consistentes y de alta calidad, lo que permitirá generar un modelo de predicción más preciso, fiable y funcional.

3. Conjunto de datos

El conjunto de datos utilizado para este reporte es el "*Titanic - Machine Learning From Disaster*", disponible en la plataforma Kaggle. Este dataset, contenido en el archivo denominado "*train.csv*", consta de un total de 891 instancias caracterizadas por 12 variables. La siguiente tabla presenta a detalle los features y labels incluidos, el tipo de dato asociado a cada una y una breve descripción de su significado.

Feature/Label	Descripción	Tipo de dato
survival (label)	Muestra si el pasajero sobrevivió (1) o no sobrevivió (0)	numérico
passengerID	ID del pasajero	numérico
pclass	Muestra la clase a la que pertenece el ticket: 1 = 1st, 2 = 2nd, 3 = 3rd	numérico
name	Muestra el nombre del pasajero	categórico
sex	Muestra el sexo del pasajero	categórico
age	Muestra la edad del pasajero (en años)	numérico
sibsp	Muestra el número de hermanos o esposas en la embarcación	numérico
parch	Muestra el número de padres o hijos en la embarcación	numérico
ticket	Muestra el número del ticket del pasajero	categórico
fare	Muestra la tarifa que pagó el pasajero para abordar	numérico
cabin	Muestra el número de cabina del pasajero	numérico
embarked	Muestra el puerto de embarque del pasajero: C = Cherbourg, Q = Queenstown, S = Southampton	categórico

Tabla 1. Descripción de los features y la etiqueta del dataset, junto con su tipo de dato.

Después de describir las características del dataset, se considera útil visualizar ejemplos reales para ilustrar cómo se estructuran los datos. A continuación, se presenta una tabla con una muestra representativa de registros que incluye valores reales para algunos features de los descritos anteriormente.

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S
2	1	1	Mrs. John Bradley	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Miss. Laina	female	26.0	0	0	STON/O 2. 3101282	7.925	NaN	S
4	1	1	Mrs. Jacques Heath	female	35.0	1	0	113803	53.1	C123	S
5	0	3	Mr. William Henry	male	35.0	0	0	373450	8.05	NaN	S

Tabla 2. Ejemplos reales del dataset

Una vez identificadas las características del dataset, se procede a evaluar la calidad e integridad de los datos. Este análisis se centra en detectar la presencia de datos faltantes, lo cual es un aspecto crítico que podría impactar negativamente en el rendimiento de los modelos que se implementarán posteriormente.

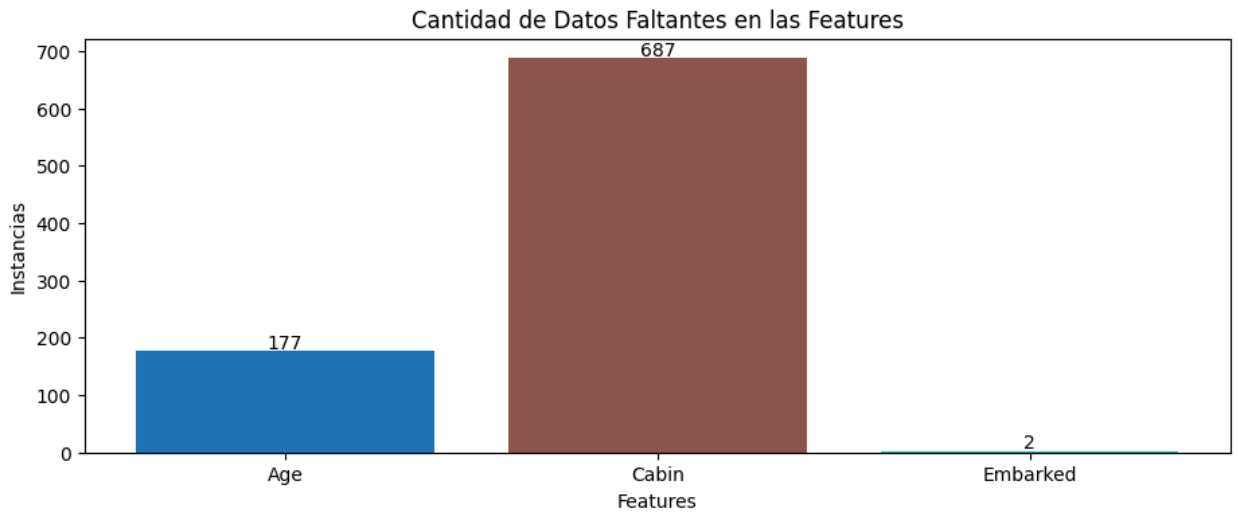

Figura 1. Distribución de datos faltantes en las features que presentan valores nulos.

En la Figura 1, se han excluido los features que no presentaron datos faltantes, quedando únicamente aquellos que sí los presentan: *Age*, *Cabin* y *Embarked*. A continuación, se muestra una tabla que detalla el número y porcentaje de datos faltantes correspondientes a las variables mencionadas.

Feature	**Número de Datos Faltantes**	**Porcentaje de Datos Faltantes**
age	177	19.87%
cabin	687	77.10%
embarked	2	0.22%

Tabla 3. Número y porcentaje de datos faltantes por feature en el dataset.

Finalmente, después de mostrar los porcentajes de datos nulos en el dataset, se considera relevante examinar la distribución de las clases en el label *survived*. La figura 2 presenta un gráfico de pastel que ilustra la proporción de registros correspondientes a los pasajeros que sobrevivieron y aquellos que no lo hicieron. La visualización proporciona una compresión clara de la representación de cada clase (0 y 1) dentro del conjunto de datos

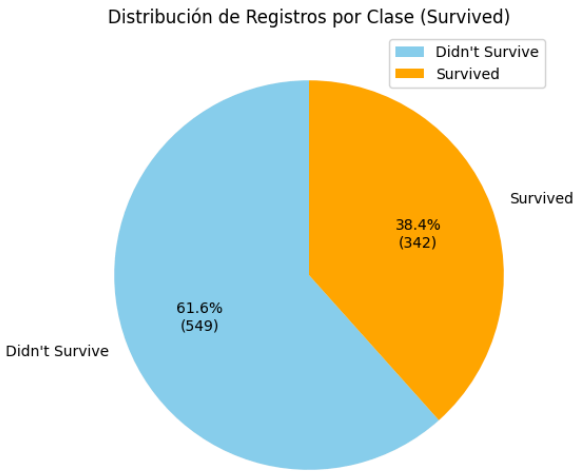

Figura 2. Distribución de registros por clase en la variable Survived.

4. Exploración y visualización de datos

La exploración y visualización de datos constituye el primer paso fundamental para comprender la estructura del dataset y las características que lo componen. En esta sección, se analizarán diversas características del dataset, con un enfoque particular en la identificación de datos faltantes, la relevancia de ciertas features y su relación con el label.

4.1 Features con datos faltantes

Esta sección examina las características del dataset que presentan datos nulos o faltantes. Según el grado de incompletitud y la relevancia de la información que aportan, algunas de estas características se eliminarán del análisis con el fin de evitar sesgos y mejorar la robustez del modelo.

4.1.1 Cabin

Cabin presenta un total de 697 valores faltantes, lo que representa el 77% del total de instancias. Esta alta proporción de datos faltantes dificulta la identificación de patrones confiables para imputar valores, lo que podría inducir ruido en el modelo.

Aunque, teóricamente, esta variable podría estar relacionada con la supervivencia debido a la ubicación de la cabina en el barco, la gran cantidad de datos faltantes hace que esta información sea poco confiable. Además la probabilidad de supervivencia también puede estar reflejada en otras variables con datos más completos. Por estas razones, se decidió eliminar la columna *cabin* para evitar comprometer la calidad del modelo.

4.1.2 Edad (Age)

La edad es un factor relevante en el análisis, ya que se encuentra fuertemente relacionado con la probabilidad de supervivencia. A continuación, se presenta un gráfico que ilustra el número de supervivientes y no supervivientes, organizados por los siguientes rangos de edad: 0-12, 13-18, 19-35, 36-50 y 51-80 años.

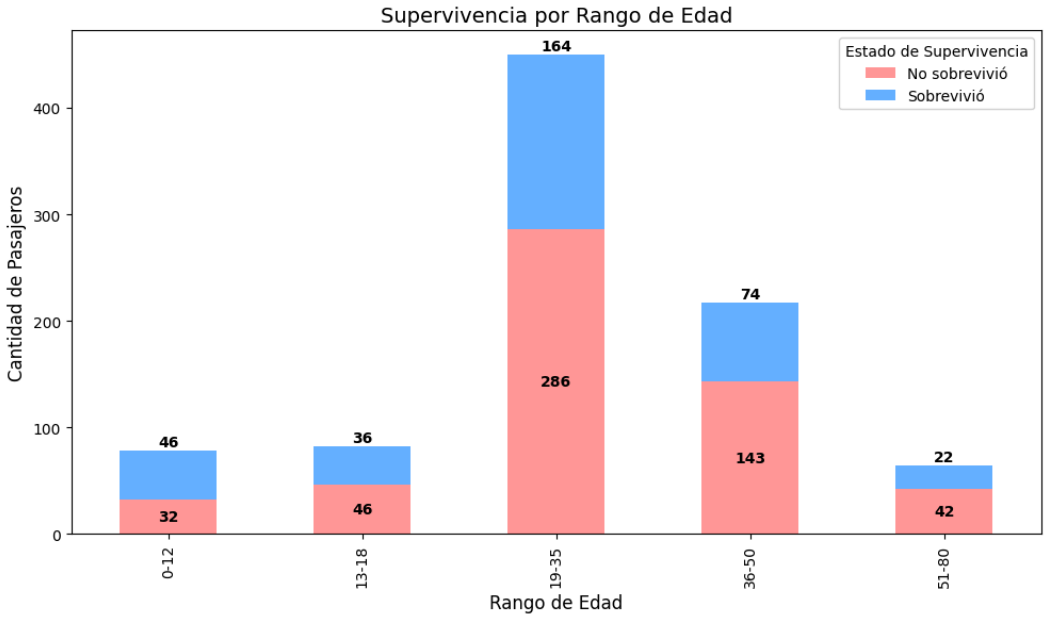

Figura 3. Distribución de supervivencia por rango de edades.

Se observa en la Figura 3 que el rango de edad entre 19 y 35 años tiene el mayor número de pasajeros, con una notable proporción de personas que no sobrevivieron. En contraste, los grupos de edad más jóvenes (0-12 años) y mayores (51-80 años) tienen una menor cantidad de pasajeros, pero presentan una mayor supervivencia, especialmente en el primer grupo. Esto sugiere que la edad es un factor influyente en la probabilidad de supervivencia, siendo los adultos jóvenes los más afectados.

4.1.3 Embarked

Embarked presenta solo dos datos faltantes, lo que representa el 0.22% de las instancias totales, lo cual llevó a evaluar su pertinencia dentro del dataset final.

Para determinar su relevancia, se analizó la relación entre las ciudades de embarque y la tasa de supervivencia, con el objetivo de identificar patrones significativos que justifiquen su inclusión en el modelo.

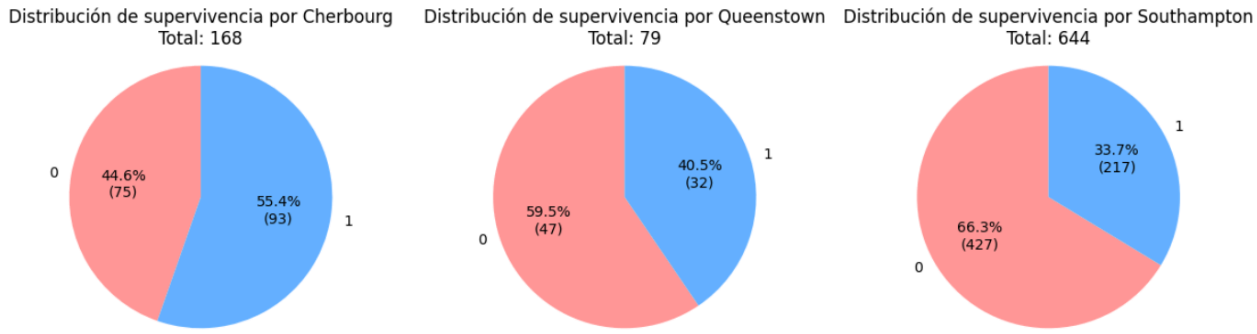

Figura 4. Distribución de supervivencia por cada puerto de embarque.

Se puede observar en la Figura 4, que la diferencia entre la tasa de supervivencia para la embarcación en *Cherbourg* es relativamente pequeña, con una proporción de 45% frente a 55%. Sin embargo, para las ciudades de *Queenstown* y *Southampton*, la proporción de supervivencia fue de aproximadamente 1 a 3, lo que significa que solo una de cada tres personas sobrevivió. Esto sugiere que la feature podría tener relevancia, especialmente considerando que la mayor parte de los pasajeros embarcaron en *Southampton* (644 de 891).

No obstante, surge la duda de si la aparente relación entre la ciudad de embarque y la tasa de supervivencia podría estar influenciada por otra variable, como el sexo. Para verificar lo anterior, se realizó la misma comparación, esta vez separando a los pasajeros por sexo, con el fin de obsevar si el patrón se repetía.

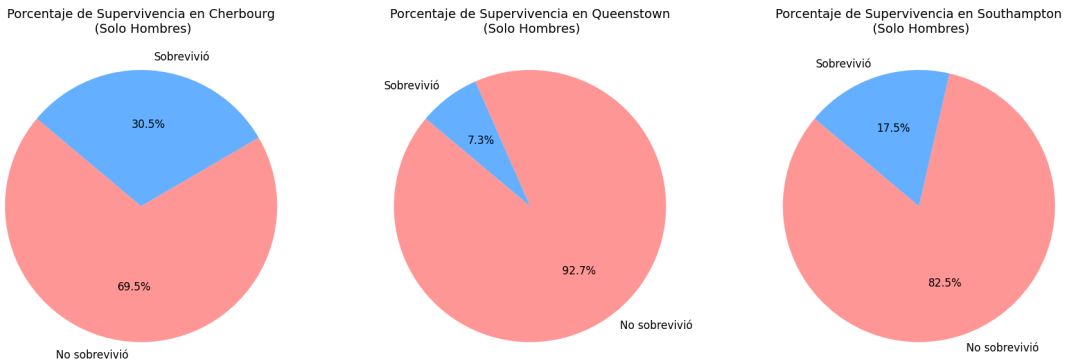

Figura 5. Comparación de *Embarked* y *Survived* para hombres.

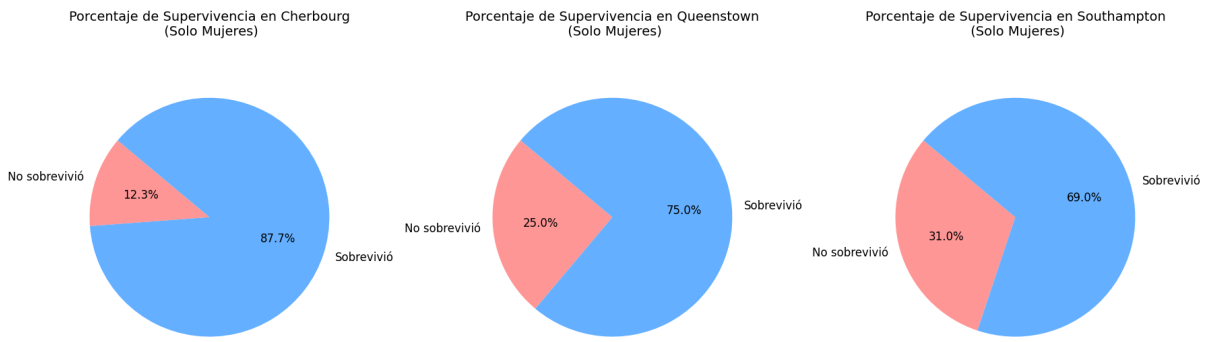

Figura 6. Comparación de *Embarked* y *Survived* para mujeres.

A partir de los gráficos anteriores, se observa que el patrón de supervivencia no es consistente cuando se desglosa por sexo, tanto en hombre como en mujeres, lo que dificulta la identificación de una relación clara y consistente entre la ciudad de embarque y la supervivencia. Aunque hay un incremento en la tasa de supervivencia en ciertos casos sugiriendo una relación, esta no es suficientemente evidente ni robusta como para considerar la eliminación del feature *Embarked*. Por lo tanto, la ambigüedad en la identificación de un patrón claro indica que es prudente mantener este feature en el análisis e implementación del modelo, debido a que podría interactuar con otras variables de manera significativa.

4.2 Features no relevantes

Esta sección examina las características del dataset que, tras un análisis inicial, se consideran no relevantes para el modelo predictivo. Dado su bajo impacto informativo y su posible contribución de ruido en el análisis, las features de esta sección serán eliminadas del dataset.

4.2.1 PassengerId

PassengerId tiene como objetivo identificar de manera única a cada pasajero dentro del dataset, sin ofrecer valor predictivo para el modelo. Por lo anterior, la variable se eliminará del dataset final.

4.2.2 Ticket

En el dataset, se identifican un total de 681 tickets únicos, aunque esta feature abarca 891 instancias en total. Un análisis más detallado reveló que los tickets duplicados corresponden a aquellos que identifican grupos de personas, generalmente familias o parejas. Por lo tanto, se concluye que esta variable cumple una función similar a *passengerId*: identificar a las personas que abordaron el Titanic. Por lo anterior, la feature de *ticket* se eliminará del dataset final.

4.2.3 Fare

Fare representa el costo del boleto y está estrechamente relacionada con *pclass*, que indica la clase del boleto. Dado que utilizar ambos datos sería redundante, se decidió conservar únicamente la columna de *pclass* y descartar la feature *fare* del dataset final.

4.3 Features relevantes

En esta sección se analizan las características del dataset que se consideran fundamentales para el modelo predictivo, cuya inclusión en el análisis es crucial para desarrollar un modelo robusto y preciso.

4.3.1 Sex

Sex es uno de los factores clave en el análisis de la supervivencia de los pasajeros del Titanic, dado que es probable que el género haya tenido un impacto significativo en las probabilidades de supervivencia. La Figura 7 muestra la distribución porcentual de supervivientes y fallecidos según el sexo.

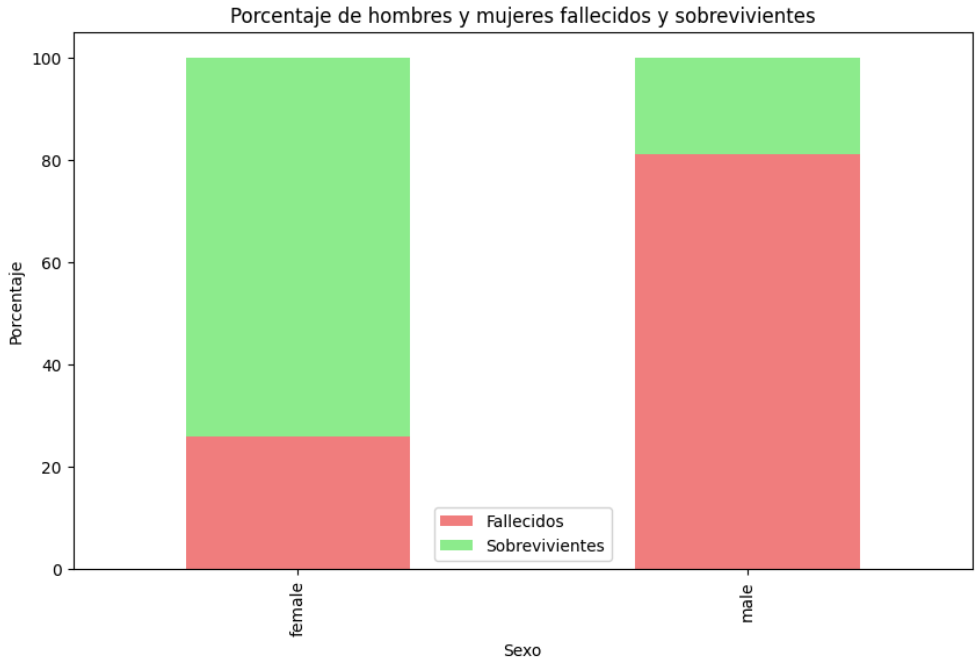

Figura 7. Distribución porcentual de supervivientes y fallecidos según el sexo.

Se puede observar que existe una diferencia notable en la probabilidad de supervivencia según el género, donde las mujeres presentaron un porcentaje de supervivencia más alto en comparación con los hombres. Esto sugiere una posible relación entre el feature *sex* y el label *survived*.

4.3.2 Pclass

Pclass refleja la categoría del boleto adquirido por el pasajero, lo que está asociado con la posición social y, quizás, el acceso a recursos adicionales durante la evacuación. La Figura 8 ilustra la relación de *pclass* con la supervivencia.

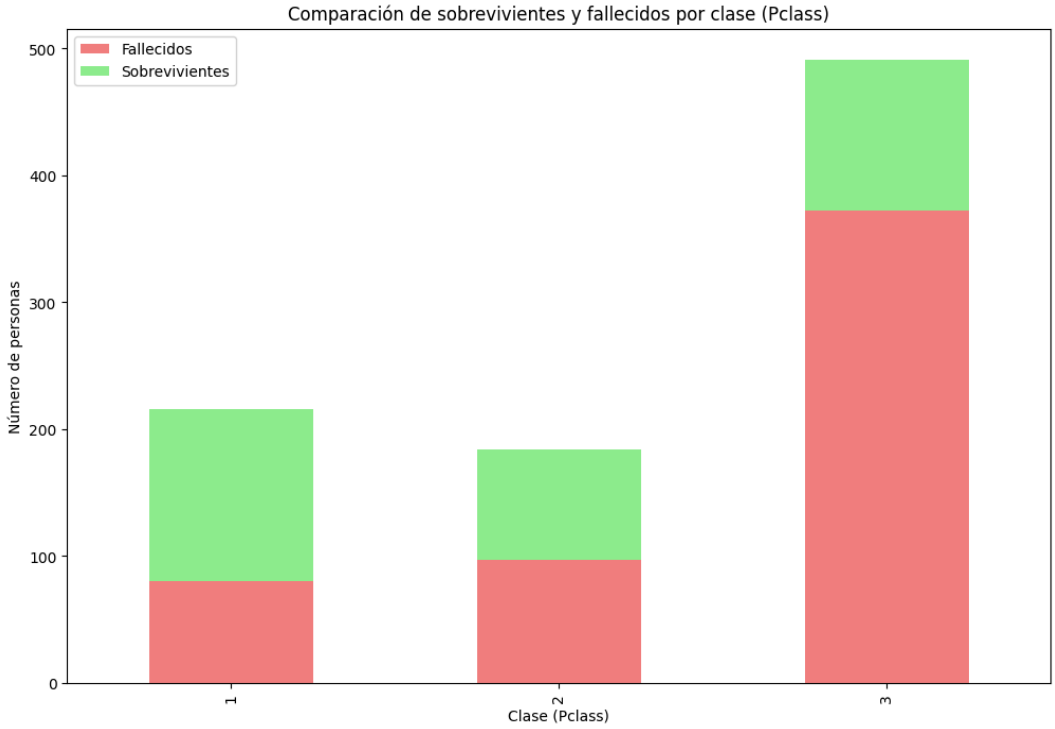

Figura 8. Comparación de supervivientes y fallecidos según la clase del boleto.

Se observa una relación entre la clase del boleto y la supervivencia, donde los pasajeros de primera clase muestran una mayor tasa de supervivencia en comparación con los de segunda y tercera clase. Sin embargo, es importante tener en cuenta que el número total de pasajeros varía entre las clases, lo que podría influir en la interpretación de los resultados.

4.3.3 Sibsp

Sibsp representa el número de hermanos o cónyuges que un pasajero tenía a bordo del Titanic. Es un factor que podría influir en la probabilidad de supervivencia debido a que la presencia de familiares podría haber afectado la evacuación de los pasajeros. Con el fin de evaluar la relevancia de esta feature en la predicción de supervivencia, se generó un gráfico que analiza la correlación entre el número de *Sibsp* abordo y el resultado de *survived* para el pasajero.

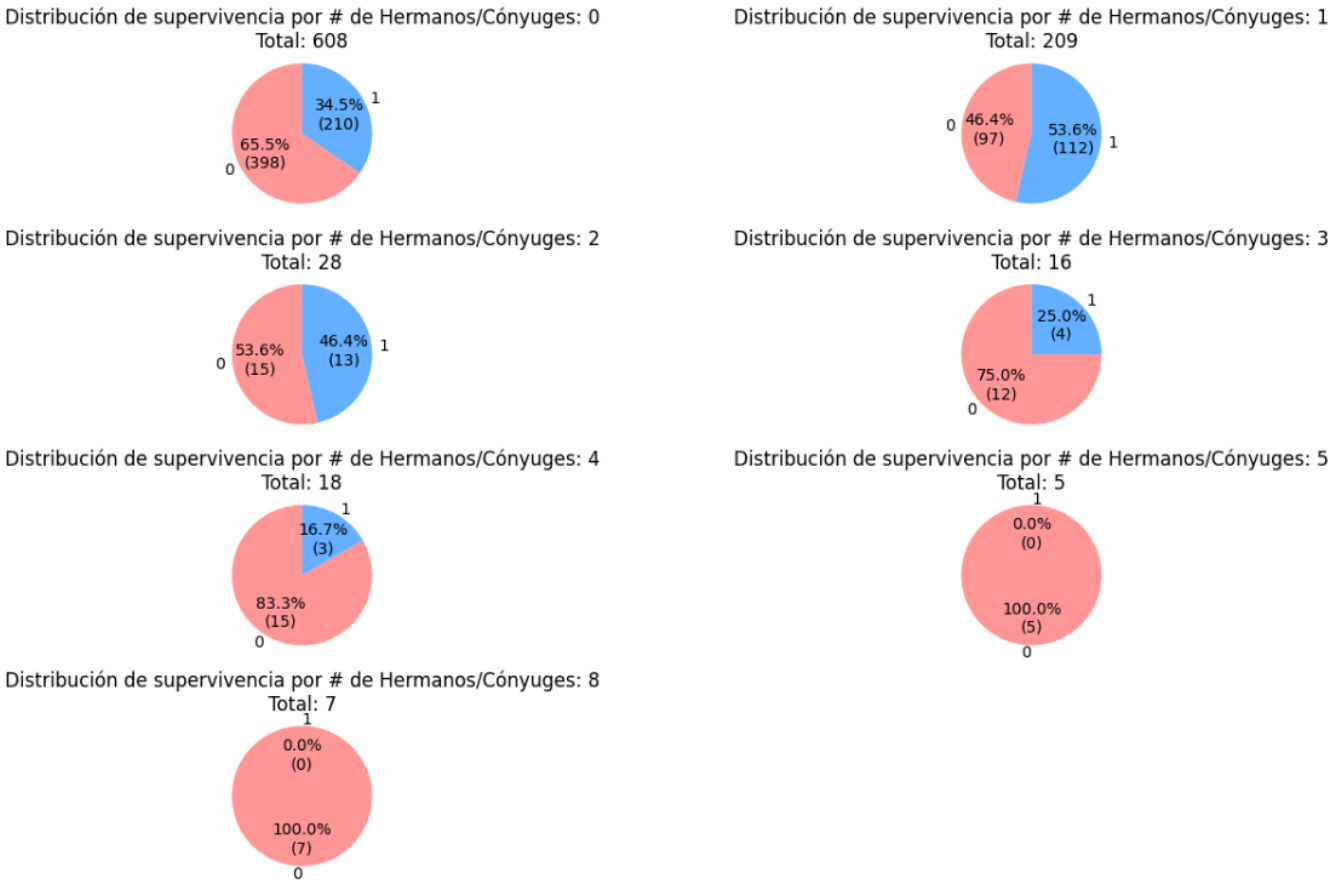

Figura 9. Distribución de supervivencia por número de hermanos / cónyuges.

Se observa en la Figura 9 que los pasajeros sin acompañantes tuvieron una tasa de supervivencia baja (34.5%). Aquellos con un solo acompañante presentaron una mayor probabilidad de supervivencia (53.6%). Sin embargo, a medida que aumenta el número de hermanos o cónyuges, la tasa de supervivencia disminuye drásticamente, especialmente para aquellos con 3 o más acompañantes, donde la tasa de supervivencia fue considerablemente más baja.

Se concluye que los pasajeros con un solo acompañante tenían una mayor probabilidad de supervivencia, mientras que aquellos con un número más elevado de acompañantes vieron disminuidas sus posibilidades de sobrevivir.

Por otra parte, con el fin de determinar si la relación observada estaba influenciada por otra variable, se aplicó la misma metodología utilizada en el análisis del feature *embarked*. Esto implicó una comparación adicional contra el sexo para verificar la consistencia de los patrones de supervivencia observados. Las figuras 10 y 11 muestran gráficamente la comparación mencionada anteriormente.

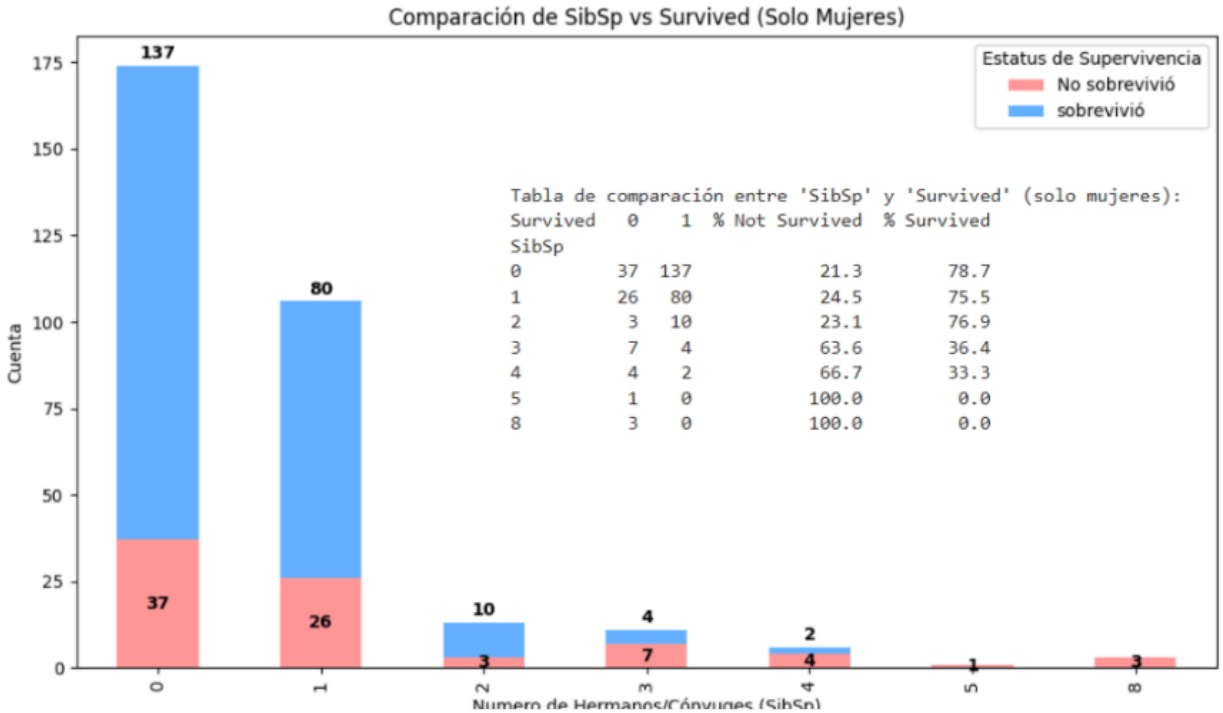

Figura 10. Comparación de *SibSp* vs *Survived* para mujeres.

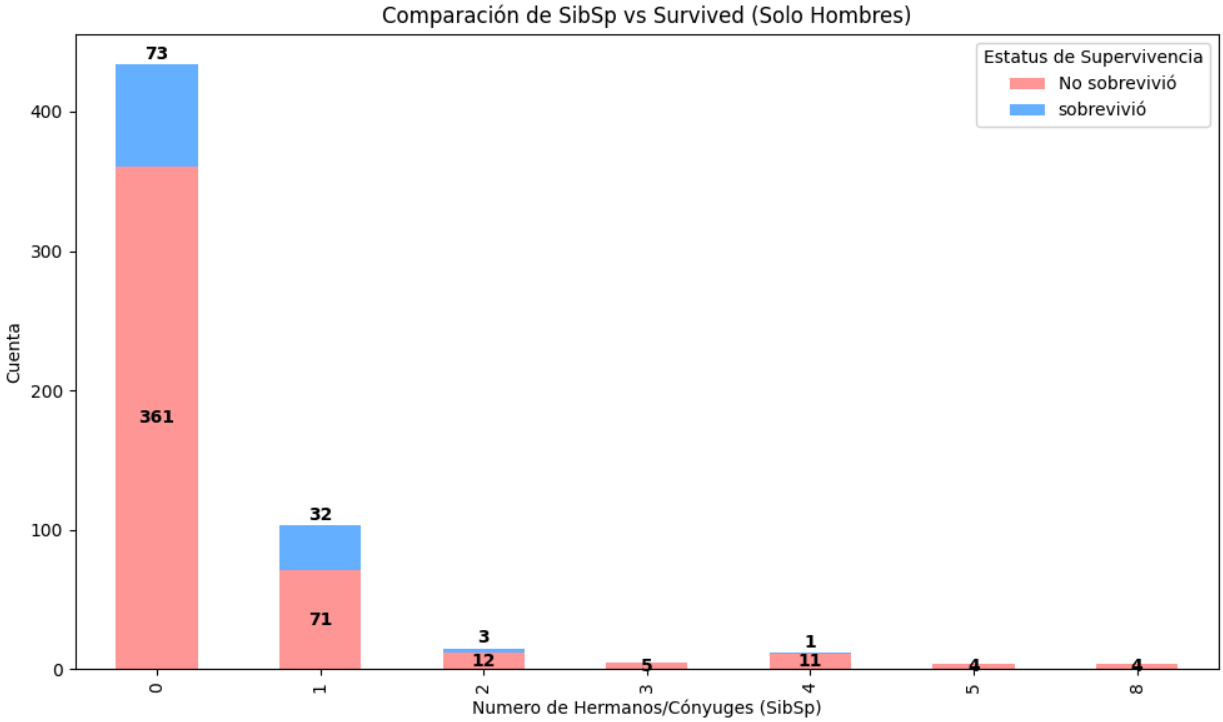

Figura 11. Comparación de *SibSp* vs *Survived* para hombres.

Los resultados obtenidos en este segundo análisis refuerzan las conclusiones derivadas de la Figura 9: a medida que disminuye el número de hermanos o cónyuges a bordo, la probabilidad de supervivencia aumenta, y viceversa. Este patrón consistente sugiere que la feature *sibsp* es un factor relevante para la predicción de la supervivencia en el problema.

4.3.4 Parch

Parch representa el número de padres e hijos que un pasajero tenía a bordo de la embarcación. Similar a *sibsp*, esta feature podría influir en la probabilidad de supervivencia debido a que la presencia de familiares podría haber afectado la toma de decisiones al momento de la evacuación. Con el fin de evaluar la relevancia de esta variable en la predicción de supervivencia, se generó un gráfico de pastel que analiza la correlación entre el número de *parch* abordo y el resultado de *survived* para el pasajero.

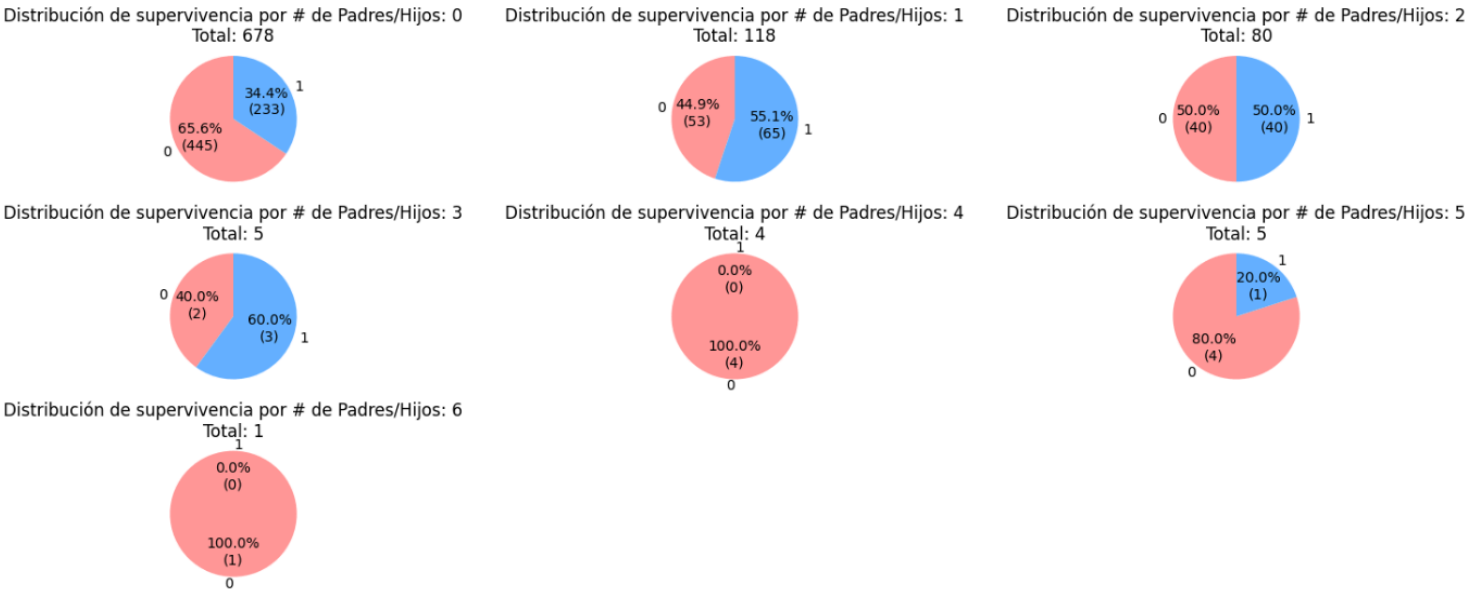

Figura 12. Distribución de supervivencia por número de padres / hijos.

Se observa en la Figura 12 que los pasajeros con menor número de *parch* tuvieron una mejor tasa de supervivencia a comparación de aquellos que contaban con un mayor número. Se concluye que los pasajeros con un uno o dos padres o hijos tenían una mayor probabilidad de supervivencia, mientras que aquellos con un número más elevado de acompañantes vieron disminuidas sus posibilidades de sobrevivir.

Por otra parte, con el fin de determinar si la relación observada estaba influenciada por otra variable, se aplicó la misma metodología utilizada en el análisis del feature *embarked*. Esto implicó una comparación adicional contra el sexo para verificar la consistencia de los patrones

de supervivencia observados. Las figuras 13 y 14 muestran gráficamente la comparación mencionada anteriormente.

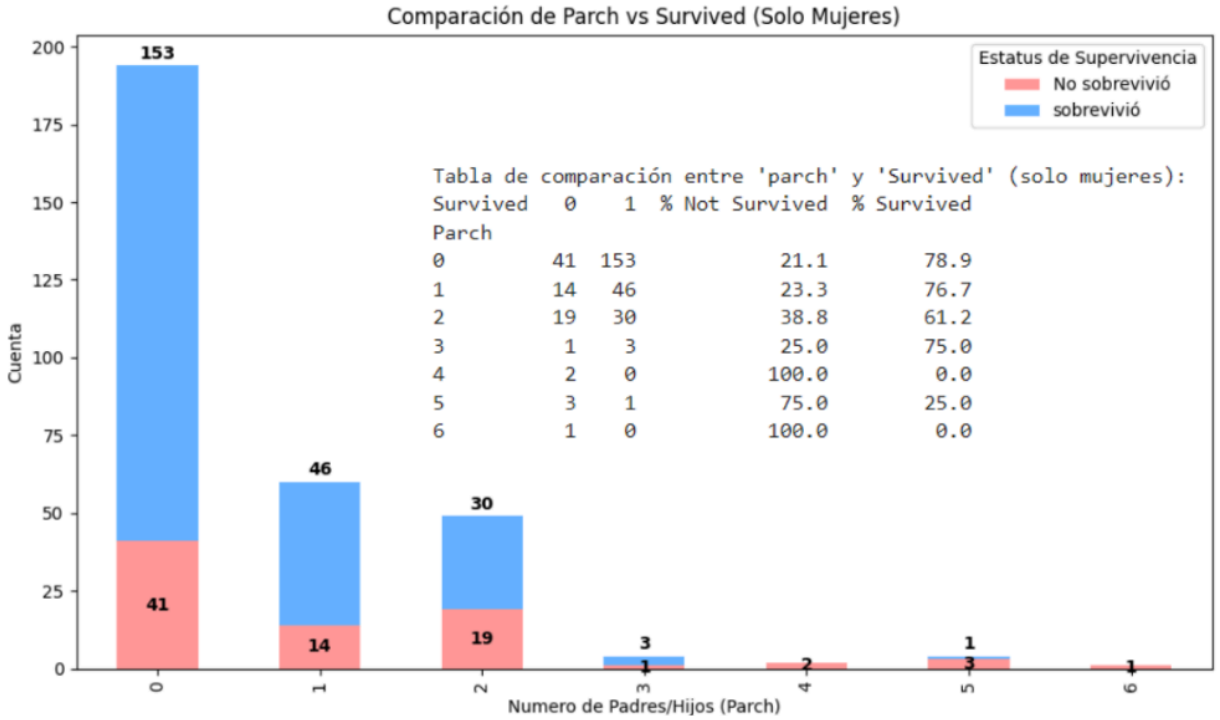

Figura 13. Comparación de Parch vs Survived para mujeres.

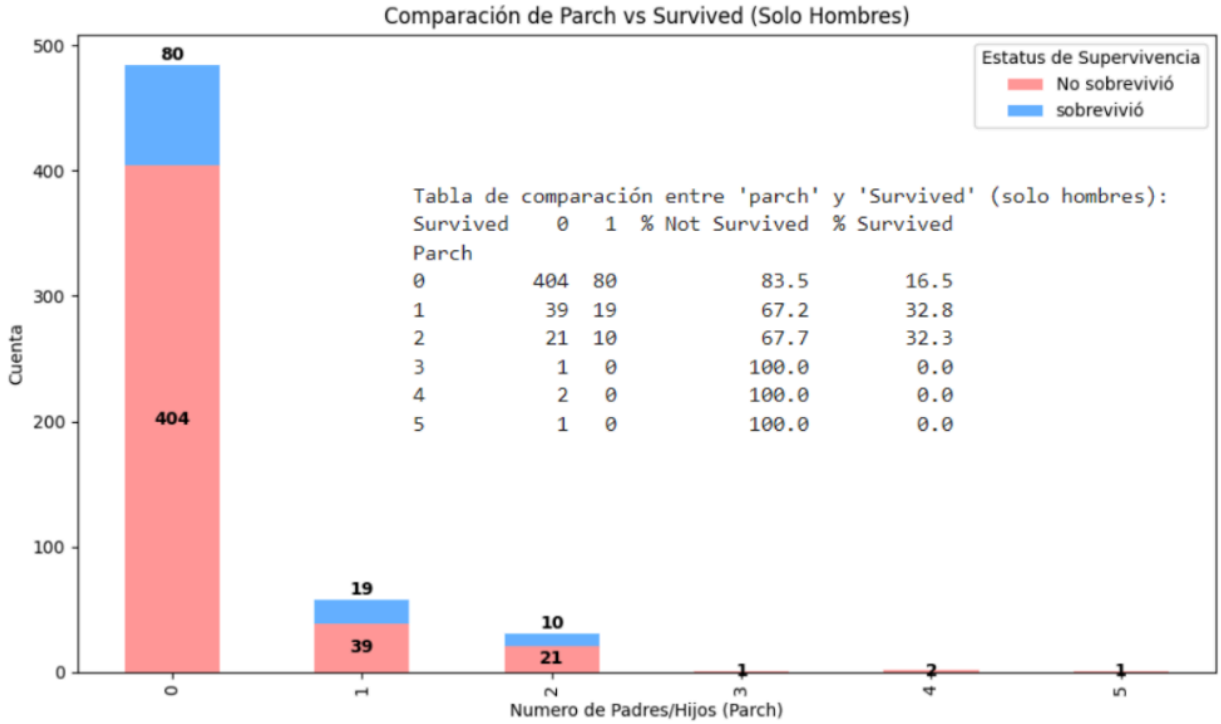

Figura 14. Comparación de Parch vs Survived para hombres.

El análisis de las figuras 13 y 14 revela que la presencia de uno o dos padres o hijos a bordo parece haber aumentado las probabilidades de supervivencia, particularmente entre las mujeres. Sin embargo, un mayor número de familiares cercanos redujo significativamente las tasas de supervivencia en ambos sexos, lo que sugiere que la feature *parch* es relevante para predecir la supervivencia.

5. Limpieza y transformación de datos

Esta sección aborda el proceso de limpieza y transformación de los datos, con el objetivo de preparar el dataset para su uso posterior con el modelo.

5.1 Edad (Age)

Como se mencionó anteriormente, la variable de *edad* presenta un total de 177 datos faltantes, lo que representa aproximadamente un 20% de las instancias. La eliminación de estas instancias podría convertirse en un problema debido a la relevancia de la feature para el análisis. Dado que la edad es considerada un factor crucial, se decidió utilizar una estrategia de imputación basada en los títulos de los pasajeros para estimar las edades faltantes. La siguiente tabla presenta un ejemplo representativo de un pasajero para cada título identificado en el dataset, adicionado de un rango de edad.

Título	Ejemplo de Dato	Rango de Edad
Capt	Capt. Edward Clifford Crosby	> 18 años
Col	Col. Oberst Alfons Simonius-Blumer	> 18 años
Don	Don. Manuel E. Uruchurtu	> 18 años
Dr	Dr. William Edward Minahan	> 18 años
Jonkheer	Jonkheer John George Reuchlin	> 18 años
Lady	Lady Lucille Duff Gordon	> 18 años
Major	Major Arthur Godfrey	> 18 años
Master	Master Gosta Leonard Palsson	< 8 años
Miss	Miss Laina Heikkinen	<= 18 años
Mlle	Mlle. Emma Sagesser	> 18 años
Mme	Mme. Leontine Pauline Aubart	> 18 años

Mr	Mr. Owen Harris Braund	> 18 años
Mrs.	Mrs. John Bradley Cummings	> 18 años
Ms	Ms. Encarnacion Reynaldo	> 18 años
Rev	Rev. Thomas Roussel Davids Byles	> 18 años
Sir	Sir. Cosmo Edmund Duff Gordon	> 18 años
Countess	The Countess Lucy Noel Rothes	> 18 años

Tabla 4. Ejemplos de datos por título y su rango de edad asociado,

Como se observa en la Tabla 4, muchos de los títulos están asociados a rangos de edad similares, lo que genera redundancia en la clasificación. Con el fin de simplificar el análisis, se decidió agrupar los títulos (y sus pasajeros) en cuatro categorías principales, cada una representando a un rango de edad distinto. Lo anterior se puede observar en la Tabla 4.

Título	**Descripción**	**Rango de Edad**
Mr.	Hombre adulto	18 < años < 70
Mrs.	Mujer casada	18 < años < 55
Miss	Mujer soltera	0 < años < 50
Master	Niño	0 < años < 8

Tabla 5. Agrupación de títulos por rango de edad,

Una vez que los pasajeros se agrupan por título, se procede a calcular el promedio y la desviación estándar de la edad para cada grupo. Estos valores proporcionan una referencia del rango típico de edades asociado a cada título. Con base en estos cálculos, se generan valores de edad dentro de este rango, utilizando la desviación estándar para mantener una distribución natural, asegurando que las edades estimadas se alineen con las características de los datos originales. Por ejemplo, si una persona con el título de *Miss* no contaba con una edad asignada, el método anterior calcularía una edad adecuada para integrarla al dataset sin afectar su integridad.

Se concluye que esta técnica de estimación presenta varias ventajas para determinar la edad utilizando la desviación estándar y un valor aleatorio dentro de ese rango. Lo anterior permite reducir el ruido en el modelo al llenar los datos faltantes sin afectar la integridad de los demás datos de la feature.

5.2 Embarked

Como solamente faltaban dos datos en 'Embarked', se decidió llenarlos con el número 2 que significa que parten del puerto de Queenstown. Esto porque de Queenstown sólo se tiene registrado que partieron 77 personas, lo cual es muy poco en comparación con los 644 y 168 personas que partieron de los otros dos puertos.

5.3 One Hot Encoding

El *one-hot encoding* es una técnica de preprocesamiento utilizada en el análisis de datos para convertir variables categóricas en un formato que puede ser fácilmente comprendido por modelos de aprendizaje automático. Consiste en representar cada categoría de una variable como una columna binaria independiente, donde un valor de "1" indica la presencia de esa categoría y un "0" indica su ausencia.

En el análisis, se aplicó *one-hot encoding* a la *feature Sex*, dividiendo esta variable en dos columnas binarias: *Male* y *Female*. De manera similar, se aplicó *one-hot encoding* a la *feature embarked*, separando los puertos de embarque en tres columnas binarias: *Cherbourg*, *Queenstown* y *Southampton*. Este enfoque permite que el modelo interprete cada categoría de manera independiente, mejorando la capacidad del modelo para analizar y predecir la supervivencia en función del género y el puerto de embarque.

6. Estructura final

El conjunto final de datos para el entrenamiento del modelo se muestra en la tabla 6, donde se presenta a detalle los features y labels finales incluidos, el tipo de dato asociado a cada una y una breve descripción de su significado.

Feature/Label	Descripción	Tipo de dato
survival (label)	Muestra si el pasajero sobrevivió (1) o no sobrevivió (0)	numérico
pclass	Muestra la clase a la que pertenece el ticket: 1 = 1st, 2 = 2nd, 3 = 3rd	numérico
sex	Muestra el sexo del pasajero	numérico
age	Muestra la edad del pasajero (en años)	numérico
sibsp	Muestra el número de hermanos o esposas en la embarcación	numérico

parch	Muestra el número de padres o hijos en la embarcación	numérico
embarked	Muestra el puerto de embarque del pasajero: 0 = Cherbourg, 1 = Queenstown, 2 = Southampton	numérico

Tabla 6. Descripción de los features y la etiqueta del dataset final, junto con su tipo de dato.

Después de describir las características del dataset final, se considera útil visualizar ejemplos reales para ilustrar cómo se estructuran los datos. A continuación, se presenta una tabla con una muestra representativa de registros que incluye valores reales para algunos features de los descritos anteriormente.

Survived	Pclass	Male	Female	Age	SibSp	Parch	Cherbourg	Queenstown	Southampton
0	3	1	0	22.0	1	0	0	0	1
1	1	0	1	38.0	1	0	1	0	0
1	3	0	1	26.0	0	0	0	0	1
1	1	0	1	35.0	1	0	0	0	1
0	3	1	0	35.0	0	0	0	0	1

Tabla 7. Ejemplos reales del dataset final

7. Modelos

Esta sección se enfoca en la implementación y evaluación de modelos de machine learning. Se exploran diferentes enfoques de modelado, analizando sus rendimientos y comparando los resultados mediante métricos estándar. El objetivo es identificar el modelo que mejor capture las relaciones entre los features y el label del dataset final, para posteriormente elegir uno y realizar un fine-tuning que optimice su precisión y capacidad predictiva.

7.1 Modelos Utilizados

7.1.1 Regresión Logística

La regresión logística es un modelo estadístico utilizado para predecir la probabilidad de un evento binario, es decir, un resultado que puede tener dos valores posibles, en este caso, “sobrevivió / no sobrevivió”. Se basa en la relación entre una o más variables independientes y la probabilidad de que ocurra un determinado resultado.

El modelo es de gran importancia para este problema, debido a que permite modelar la relación entre las diferentes features contra la probabilidad de supervivencia y, dado que el objetivo es predecir si sobrevivió o no, este modelo es particularmente adecuado para proporcionar una estimación directa de la probabilidad de supervivencia.

7.1.2 Random Forest

Random Forest es un modelo de machine learning basado en un conjunto de árboles de decisión. El modelo crea múltiples árboles durante el entrenamiento y luego promedia sus predicciones o utiliza el voto mayoritario para tomar la decisión final. Cada árbol se entrena con una muestra aleatoria del conjunto de datos y utiliza un subconjunto aleatorio de features para dividir los nodos, lo cual mejora la precisión de las predicciones evitando el overfitting.

Para este problema, Random Forest es una técnica pertinente ya que permite manejar un conjunto de datos con múltiples características, algunas de las cuales pueden ser altamente correlacionadas o no lineales.

7.1.3 Modelo K-Nearest Neighbors (KNN)

El modelo *k*-Nearest Neighbors (KNN) es un algoritmo de aprendizaje supervisado utilizado tanto para clasificación como para regresión. Su funcionamiento se basa en la idea de que los datos similares se encuentran cerca unos de otros en el espacio de características. El modelo predice la clase de una nueva instancia de datos en función de las clases de sus *k* vecinos más cercanos en el conjunto de entrenamiento. En la etapa de clasificación, asigna una etiqueta a una instancia desconocida basándose en la mayoría de las etiquetas de los *k* vecinos más cercanos. Para la regresión, el modelo predice un valor basado en el promedio de los valores de los *k* vecinos más cercanos.

KNN puede ser una herramienta valiosa ya que tras el preprocesamiento y la limpieza de datos, incluyendo la transformación de variables categóricas y la gestión de valores faltantes, el modelo permitirá clasificar a los pasajeros en función de sus características y predecir la probabilidad de supervivencia. Además, no asume una forma específica para la relación entre las características y

la variable objetivo, lo que lo hace flexible y adaptativo para la clasificación de supervivencia en el Titanic.

7.1.4 Red Neuronal

Finalmente, la red neuronal. Este es un modelo de machine learning el cual está compuesto por capas de neuronas que procesan la información recibida por la capa anterior y la transmite a la siguiente.

Para este problema en particular, una modelo de este tipo es particularmente adecuado debido a su capacidad para capturar y modelar relaciones complejas y no lineales entre múltiples features, como la edad, el sexo o la clase del boleto, los cuales influyen en la probabilidad de supervivencia. Estas características permiten que el modelo aprenda patrones ocultos en los datos, optimizando la precisión en la predicción de la supervivencia de los pasajeros

7.2 Configuración de los Modelos

7.2.1 Configuración Regresión Logística

Para desarrollar la regresión logística se optimizaron tres principales hiperparámetros, que son los siguientes:

- **Learning Rate (Alpha):** Este parámetro determina el tamaño de los pasos que da el algoritmo durante el proceso de optimización. Un valor demasiado alto puede provocar una convergencia rápida o inestabilidad en el modelo, mientras que uno demasiado bajo puede ralentizar el aprendizaje.

- **Epochs**: Representa el número de veces que el algoritmo recorre el conjunto de datos completo durante el entrenamiento. Un mayor número de epochs puede mejorar el aprendizaje del modelo, aunque un número excesivo podría resultar en overfitting.

- **Umbral (Threshold):** En regresión logística, el umbral define el punto a partir del cual la probabilidad predicha se clasifica como una clase positiva (por ejemplo, 1 en lugar de 0). Si la probabilidad es mayor o igual al umbral, se clasifica como 1; de lo contrario, se clasifica como 0.

En la tabla 8 se observa que el número de epochs y el umbral varían considerablemente entre las distintas configuraciones, lo que puede parecer aleatorio. Esto se debe a que, aunque el número de epochs está limitado a diez mil, el entrenamiento se detiene automáticamente cuando la diferencia en el costo entre el epoch actual y el anterior es menor a 0.0001, ya que cualquier mejora posterior sería mínima.

Respecto al umbral, debido al desbalance en el dataset, con una proporción de 60:40 entre No Survived y Survived, se optó por no utilizar el umbral estándar de 0.5. En su lugar, se probaron diversos valores entre 0.2 y 0.9 en intervalos de 0.01 para encontrar el valor óptimo para cada configuración entrenada.

ID	Alpha	Epochs	Umbral
1	0.01	603	0.42
2	0.1	20	0.38
3	0.25	224	0.67
4	0.5	143	0.66
5	1	61	0.68

Tabla 8. Configuraciones de Regresión Logística

7.2.2 Configuración Random Forest

Para desarrollar el modelo de Random Forest, se probaron y optimizaron varios hiperparámetros clave mediante una búsqueda exhaustiva utilizando GridSearchCV. Los principales parámetros evaluados y su impacto en el rendimiento del modelo fueron los siguientes:

- **Número de árboles (n_estimators):** Este parámetro determina la cantidad de árboles en el bosque. Un mayor número de árboles suele mejorar la precisión del modelo, aunque incrementa el tiempo de entrenamiento.

- **Máxima profundidad del árbol (max_depth):** Controla la profundidad máxima que puede alcanzar cada árbol. Aunque una mayor profundidad permite capturar más patrones en los datos, también aumenta el riesgo de overfitting en el conjunto de entrenamiento.

- **Mínimo de muestras para dividir un nodo (min_samples_split):** Define el número mínimo de muestras requeridas para dividir un nodo. Valores más altos tienden a hacer que los árboles sean más generales, lo que puede ayudar a prevenir el overfitting.

- **Mínimo de muestras en una hoja (min_samples_leaf):** Establece el número mínimo de muestras necesarias en una hoja o nodo terminal. Ajustar este parámetro ayuda a suavizar el modelo, especialmente en conjuntos de datos pequeños.

- **Bootstrap:** Este parámetro indica si las muestras de los datos se toman con reemplazo o sin él al construir cada árbol. El uso del bootstrap puede mejorar la robustez del modelo al introducir variabilidad en las muestras utilizadas para la construcción de cada árbol.

La tabla 9 muestra las mejores combinaciones de hiperparámetros encontrados mediante este proceso de optimización.

ID	n_estimators	max_depth	min_samples_split	min_samples_leaf	bootstrap
1	50	20	4	5	True
2	100	10	4	2	True
3	100	None	4	2	True
4	200	None	4	10	True
5	200	20	4	10	True

Tabla 9. Configuraciones de Random Forest

7.2.3 Configuración del K-Nearest Neighbors (KNN)

Para optimizar el rendimiento del modelo, se evaluaron las siguiente combinaciones de hiperparámetros:

- **Número de Vecinos (n_neighbors):** Con los valores de 3, 5, 7, 9, y 11.

- **Ponderación de Vecinos (weights):** Se evaluaron dos opciones: uniform, donde todos los vecinos tienen el mismo peso y distance, donde los vecinos más cercanos tienen un peso mayor.

- **Métrica de Distancia (metric):** Se analizaron tres métricas: euclidean, manhattan, y minkowski.

- **Algoritmo de Búsqueda:** Se utilizaron los algoritmos auto, ball_tree, kd_tree, y brute para la búsqueda de los vecinos más cercanos.

La tabla 10 presenta las configuraciones más relevantes según los resultados obtenidos (7.3.4).

ID	n_neigbors	weights	metric	algorithm
1	7	uniform	manhattan	ball_tree
2	11	uniform	manhattan	ball_tree
3	9	uniform	euclidean	brute
4	9	uniform	minkowski	brute
5	7	uniform	manhattan	kd_tree
6	7	uniform	manhattan	auto

Tabla 10. Configuraciones de la K-Nearest Neighbors

7.2.4 Configuración de la Red Neuronal

Para la implementación de la red neuronal, se utilizaron diversas técnicas con el objetivo de mejorar tanto la estabilidad del entrenamiento como la capacidad de generalización del modelo. Las configuraciones críticas de la red neuronal se describen a continuación:

- **Dropout:** Consiste en desactivar aleatoriamente un porcentaje de las neuronas durante cada epoch del entrenamiento, acción que ayuda a prevenir el overfitting al asegurar que el modelo no dependa en exceso de neuronas específicas, permitiendo una mejor generalización cuando se expone a datos nuevos.

- **Batch Normalization:** Estandariza los outputs y estabiliza el proceso de aprendizaje al mitigar problemas relacionados con la inicialización de los pesos, mejorando la generalización del modelo.

- **Early Stopping:** Detiene el entrenamiento cuando el rendimiento del modelo en el test set deja de mejorar, evitando el overfitting.

- **Learning Rate Reduction:** Ajusta dinámicamente el learning rate cuando una métrica no ha mejorado en un número determinado de epochs (*patience*), lo cual permite afinar el modelo con mayor precisión durante la etapa de entrenamiento.

- **Función de Activación:** Se implementó la función de activación ReLU en las capas ocultas de la red y Sigmoid para la capa de salida, debido a que el label es de naturaleza binaria (0 ó 1).

- **Batch Size:** Ajustado para equilibrar la velocidad del entrenamiento y la precisión del modelo. Cuando esta configuración es más grande, se obtienen actualizaciones más estables en los pesos. En cambio, si es pequeño puede generar ruido en el entrenamiento, pero también puede terminar en una mejor generalización.

La siguiente tabla muestra cinco distintas configuraciones que se aplicaron a la red neuronal.

ID	Neuronas	Dropout	Patience	α	Epochs	Batch_Size
1	[64, 128, 128, D, 64, 32, B, 32, 64, D, 128, 128, 1]	0.3	200	0.00001	2000	180
2	[64, 128, 128, D, 64, 32, B, 32, 64, D, 128, 128, 1]	0.3	200	0.00001	2000	256
3	[64, 128, 128, D, 64, 32, B, 32, 64, D, 128, 128, 1]	0.3	200	0.0001	2000	256
4	[64, 128, 128, D, 64, 32, B, 32, 64, D, 128, 128, 1]	0.3	200	0.0001	2000	512
5	[64, 128, 128, D, 64, 32, B, 32, 64, D, 128, 128, 1]	0.3	200	0.0001	2000	256

Tabla 11. Configuraciones de la red neuronal

7.3 Métricas de los Modelos

7.3.1 Métricas de la Regresión Logística

De acuerdo con los ajustes descritos en la sección 7.2.2, este apartado presenta los resultados del modelo final, evaluados mediante las métricas de Precision, Recall, F1 Score y Accuracy. Es importante señalar que, aunque el costo no se utilizó como una medida directa del rendimiento, se consideró como una herramienta valiosa para optimizar el proceso de entrenamiento, especialmente para determinar el número ideal de epochs. Los resultados obtenidos se resumen en la tabla 12.

ID	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
1	55	80	65	69
2	76	59	67	77
3	87	71	78	85
4	87	71	78	85
5	82	70	75	83

Tabla 12. Métricas de Regresión Logística

Al examinar los resultados presentados en la Tabla 12, se observa que los modelos tres y cuatro tienen un rendimiento equivalente en todas las métricas evaluadas. Para identificar el más adecuado, es necesario comparar sus configuraciones. Dado que ambos enfoques logran una generalización efectiva, el criterio clave debe ser la eficiencia en el uso de recursos durante el entrenamiento. En este sentido, el cuarto esquema sobresale, ya que alcanzó su rendimiento óptimo en solo 143 epochs, mientras que el tercero necesitó 224 epochs para obtener resultados similares.

En resumen, el cuarto modelo no solo ofrece un rendimiento superior en las métricas, sino que también muestra una mayor eficiencia en términos de recursos, superando al tercero al lograr un desempeño comparable con un menor costo computacional.

7.3.2 Métricas del Random Forest

Según lo detallado en la sección 7.2.2, a continuación se presentan los resultados obtenidos para el modelo Random Forest, evaluados a través de la métrica de Accuracy. Esta métrica se empleó para medir el rendimiento del modelo durante el proceso de entrenamiento y para identificar la configuración que proporcionó el mejor equilibrio entre precisión y capacidad de generalización.

ID	Accuracy (%)
1	82.44
2	82.16
3	82.02
4	82.02
5	82.02

Tabla 13. Métricas de Random Forest

De acuerdo con las configuraciones descritas en la sección 7.2.2 y los resultados mostrados en la tabla 13, se observa una notable similitud en el valor de accuracy entre los cinco principales algoritmos. No obstante, el primer modelo destaca ligeramente con un accuracy de 82.44%, superando al segundo por un 0.28%. Aunque esta diferencia es mínima, resalta la precisión de la primera configuración, haciéndola la más eficiente en términos de rendimiento.

Además, considerando que el parámetro n_estimators tiene un impacto considerable en el tiempo de entrenamiento, es crucial elegir la configuración que optimice el uso de recursos. En este sentido, la primera configuración se destaca por su eficiencia. Por lo tanto, la mejor configuración para el algoritmo Random Forest es la primera.

7.3.3 Métricas del K-Nearest Neighbors (KNN)

Utilizando las configuraciones detalladas en la sección 7.2.4, se presenta la siguiente tabla que muestra las métricas del rendimiento del modelo en términos de accuracy, recall y la puntuación f1.

ID	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
1	83.47	83.24	82.95	83.24
2	83.64	83.24	82.88	83.24
3	82.71	82.68	82.47	82.68
4	82.71	82.68	82.47	82.68
5	82.96	82.68	82.35	82.68
6	82.96	82.68	82.35	82.68

Tabla 14. Métricas de K-Nearest Neighbors

El análisis de las métricas mostró que la configuración 1, con n_neighbors = 7, métrica Manhattan y el algoritmo ball tree, alcanzó el mayor accuracy (83.24%) sin comprometer otras métricas. Se observó que con menos vecinos (3 y 5) el modelo era más sensible al ruido, afectando su capacidad de generalización. Con más vecinos (7, 9, 11), el modelo se estabilizó, destacando el uso de 7 vecinos como la mejor opción.

Además, la ponderación uniforme de los vecinos, combinada con la métrica Manhattan, demostró ser más efectiva que la ponderación por distancia. Aunque otras configuraciones como la métrica euclidiana con brute_force también fueron competitivas, la configuración 1 mostró un rendimiento superior en términos de precisión, estabilidad, recall y f1score, por lo que se utilizará como referencia para compararla con los otros modelos.

7.3.4 Métricas de la Red Neuronal

Utilizando las configuraciones detalladas en la sección 7.2.1, a continuación se presenta una tabla que muestra las métricas de rendimiento asociadas a cada configuración. Estas métricas incluyen la pérdida (*loss*) al comparar con el test set, así como la precisión (*accuracy*) obtenida en el mismo conjunto de pruebas.

ID	Pérdida en Test	Accuracy en Test (%)
1	0.54	75.56
2	0.4855	80.00
3	0.5682	73.33
4	0.5307	78.88
5	0.6561	72.22

Tabla 15. Métricas de rendimiento por configuración de la red neuronal

A partir de las métricas presentadas, se observa que la configuración número 2 obtuvo el mejor rendimiento en términos de precisión (80%) y la menor pérdida (0.4855) al comparar con el test set. Lo anterior indica que esta configuración logra un equilibrio adecuado entre la complejidad del modelo y su capacidad de generalización, lo que le permite hacer predicciones más precisas sobre datos nuevos.

Por otro lado, la configuración número 5 mostró el peor rendimiento, con una precisión de 72.22% y una pérdida más alta de 0.6561, sugiriendo que el modelo no está bien ajustado o que sufrió de alta varianza o bias en comparación con las demás comparaciones.

En conclusión, la configuración número 2 de la red neuronal será la elegida para ser comparada con los demás modelos, dado su rendimiento superior en términos de precisión y pérdida.

7.4 Comparación de Rendimiento

En esta sección, se comparan los rendimientos de los modelos probados para determinar cuál ofrece la mejor combinación de precisión, capacidad de generalización, y eficiencia computacional. Se evaluaron cuatro modelos principales: Regresión Logística, Random Forest, K-Nearest Neighbors (KNN) y una Red Neuronal. La comparación se basa en las métricas de precisión (accuracy), F1-Score, pérdida (loss), y la capacidad del modelo para generalizar en datos no vistos previamente.

Modelo	Pérdida en Test	Accuracy en Test (%)	F1-Score (%)
Regresión Logística	0.9219	85	78
Random Forest	N/A	82.44	N/A
KNN	N/A	83.24	82.95
Red Neuronal	0.4855	80	N/A

Tabla 16. Métricas de rendimiento de los modelos probados

Al analizar los resultados de la Tabla 16, se observa que la Regresión Logística presenta la mayor precisión con un 85%. Sin embargo, su F1-Score es de sólo 78%, lo que sugiere un desequilibrio entre la precisión y el recall. Esto es particularmente relevante considerando que, aunque los datos del dataset no están completamente balanceados, tampoco están mal distribuidos; no presentan una proporción perfecta de 50/50, pero sí una suficiente para que el F1-Score sea un indicativo clave del rendimiento del modelo.

El modelo K-Nearest Neighbors (KNN) mostró una precisión ligeramente inferior de 83.24%, pero su F1-Score fue de 82.95%, lo que indica un mejor equilibrio entre la precisión y el recall. Esta combinación de métricas sugiere que KNN es más efectivo para clasificar correctamente tanto las instancias positivas como negativas, minimizando errores en ambas direcciones. Random Forest y la Red Neuronal también mostraron buenos resultados sólidos, pero no tan sólidos como KNN o Regresión Logística.

Después de considerar los diferentes enfoques, se selecciona el modelo KNN como el más adecuado para esta tarea. Aunque la Regresión Logística presentó la mayor precisión (85%), su F1-Score más bajo (78%) refleja que el modelo podría no estar optimizado para manejar bien el equilibrio entre precisión y recall, lo cual es crucial dado que los datos del dataset no están perfectamente balanceados. Por otro lado, KNN, con una precisión del 83.24% y un F1-Score de 82.95%, ofrece un equilibrio superior, siendo capaz de generalizar de manera más consistente y

efectiva. Este rendimiento equilibrado lo convierte en la opción más robusta y confiable para predecir la supervivencia de los pasajeros del Titanic.

8. Comentarios Finales

En resumen, este estudio ha demostrado la eficacia de utilizar K-Nearest Neighbors (KNN) para predecir la supervivencia de los pasajeros del Titanic. La capacidad de KNN para capturar patrones y ofrecer resultados consistentes lo convierte en la opción más adecuada entre los modelos evaluados. Sin embargo, es importante mencionar que otro modelo, como la Regresión Logística, también ha mostrado potencial y podría ser considerado en escenarios diferentes o con un ajuste adicional. Futuros trabajos podrían centrarse en la optimización de este modelo, explorando técnicas avanzadas de regularización y selección de características para mejorar aún más la precisión y robustez de las predicciones. Este análisis sienta una base sólida para el desarrollo de sistemas predictivos en contextos similares, destacando la importancia de un enfoque integral que combine tanto la capacidad de predicción como la simplicidad en la implementación.

9. Referencias

Barrera, P. (2022, June 9). *Las probabilidades de sobrevivir en el Titanic, según la Ciencia de los Datos*. Actualidad UVG. Retrieved August 27, 2024, from https://noticias.uvg.edu.gt/probabilidades-sobrevivir-titanic-ciencia-de-los-datos-power-bi/

Brewster, H., & Coulter, L. (1999). *Tout ce que vous avez toujours voulu savoir sur le "Titanic"*. Ed. Glénat.

Cukierski, W. (2012). *Titanic - Machine Learning from Disaster*. Kaggle. Retrieved August 27, 2024, from https://www.kaggle.com/competitions/titanic