Tecnológico de Monterrey

INTELIGENCIA ARTIFICIAL AVANZADA
PARA LA CIENCIA DE DATOS I

TC3006C

Reto: Titanic - Machine Learning from Disaster

Autores:
Daniela Jiménez Téllez - A01654798
Lautaro Gabriel Coteja - A01571214
Andrés Villarreal González - A00833915
Héctor Hibran Tapia Fernández - A01661114

Índice

1. Introducción

El hundimiento del RMS (Royal Mail Ship) Titanic, uno de los barcos más importantes en su momento, ha sido una de las tragedias más tristes y habladas a lo largo de la historia.

El objetivo de este proyecto es realizar el reto propuesto por Kaggle: *Titanic - Machine Learning from Disaster*. Este consisiste en usar técnicas de Machine Learning para crear un modelo que prediga qué pasajeros tienen más posibilidad de sobrevivir la colisión del Titanic dependiendo de su nombre, edad, género, clase de ticket, entre otras características.

Habiendo dicho esto, en este reporte se muestra la resolución de este reto, junto con el procesamiento y análisis de datos, así como los resultados usando diferentes algoritmos de Machine Learning, como lo son Redes Neuronales, Random Forest y Regresión Logística.

2. Marco Teórico

El RMS Titanic, considerado uno de los barcos más grandes y lujosos de su época, estaba dividido en diferentes secciones que reflejaban no solo su estructura física, sino también la jerarquía social de los pasajeros que viajaban a bordo. A continuación, se describe cómo estaba dividido el Titanic y las ubicaciones de las cabinas según la clase social:

2.1. División del Titanic

El Titanic estaba organizado en 10 cubiertas principales, cada una con diferentes funciones y áreas dedicadas a pasajeros, tripulación y maquinaria. Estas cubiertas, de arriba hacia abajo, eran las siguientes:

- **Cubierta A (Boat Deck):** La más alta del barco, albergaba los botes salvavidas y tenía acceso a la Primera Clase. En esta cubierta se encontraban la Gran Escalinata, el gimnasio y el Paseo Promenade.
- **Cubierta B (Promenade Deck):** Principalmente destinada a pasajeros de Primera Clase, aquí estaban las suites más lujosas, el Salón de Primera Clase, y el café Parisien.

- **Cubierta C (Bridge Deck):** Contenía las cabinas de Primera Clase y algunas cabinas de la tripulación. También se encontraba el Puente de Mando del barco.
- **Cubierta D (Saloon Deck):** Aquí se ubicaban el Gran Comedor de Primera Clase y el Comedor de Segunda Clase.
- **Cubierta E (Upper Deck):** Albergaba cabinas de Primera, Segunda y Tercera Clase, además de algunas áreas de servicio para la tripulación.
- **Cubierta F (Middle Deck):** Similar a la cubierta E, esta también contenía cabinas para las tres clases de pasajeros, así como la cocina y otras áreas de servicio.
- **Cubierta G (Lower Deck):** Principalmente dedicada a los pasajeros de Tercera Clase y áreas de servicio.
- **Cubiertas Orlop y de Carga:** Las dos cubiertas más bajas del Titanic, donde se almacenaban la carga, el correo, y las bodegas. En estas áreas también se encontraban las calderas y la sala de máquinas.

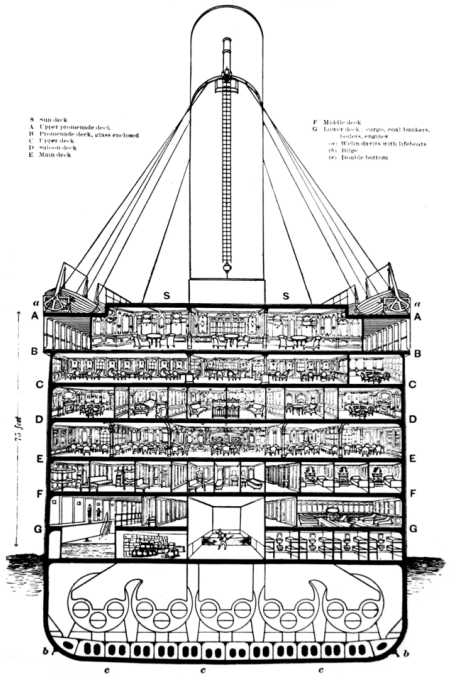

Figura 1: Representación de la división del Titanic

2.2. Ubicaciones de las Cabinas por Clase Social

El Titanic reflejaba las divisiones sociales de la época, con claras diferencias en la ubicación y las comodidades de las cabinas según la clase del pasajero.

- **Primera Clase:** Los pasajeros de Primera Clase gozaban de las mejores ubicaciones en las cubiertas superiores (A, B y C). Estas cabinas eran espaciosas y algunas de las más lujosas se encontraban en la Cubierta B, donde había suites privadas con acceso a áreas exclusivas como el Paseo Promenade.
- **Segunda Clase:** Las cabinas de Segunda Clase estaban situadas principalmente en las cubiertas D y E. Aunque no tan opulentas como las de Primera Clase, eran considerablemente más cómodas que las de Tercera Clase.
- **Tercera Clase:** Los pasajeros de Tercera Clase, compuestos en su mayoría por inmigrantes, estaban situados en las cubiertas inferiores (F y G), cerca de las áreas de servicio y las salas de máquinas. Sus cabinas eran pequeñas y compartidas por varias personas.

2.3. Regresión Logística

La Regresion Logistica es un algoritmo de clasificación que se utiliza para modelar la probabilidad de que una variable dependiente binaria (en este caso, supervivencia o no) se relacione con una o más variables independientes.

Datos que suelen usar:
Se utiliza comunmente con variables independientes categóricas y continuas. Para el reto del Titanic, datos como la clase del pasajero, su sexo, edad, y tarifa, son ejemplos típicos de variables usadas en una regresión logística.

Ventajas:

- Facil de interpretar y de implementar.
- Requiere menos computación que otros algoritmos más complejos.
- Funciona bien cuando la relación entre las variables es lineal.

Desventajas:

- No captura relaciones no lineales a menos que se introduzcan términos polinomiales o de interacción.

- Puede ser menos precisa si las variables independientes no tienen una relación clara con la variable objetivo.

2.4. Random Forest

El modelo de Random Forest es un algoritmo de clasificación y/o regresión que utiliza múltiples arboles de decisión para obtener predicciones precisas. Combina los resultados de varios arboles de decisión individuales entrenados en diferente subconjuntos del conjunto de datos.

Datos que suelen usar:
Este algoritmo puede trabajar tanto con datos categóricos como numéricos. Para el Titanic, se puede utilizar variables como la clase del pasajero, la tarifa, la edad, el puerto de embarque, etc.

Ventajas:

- Menos riesgo de sobreajuste en comparación con un solo árbol de decisión.
- Funciona bien con datos faltantes y se adapta a datasets mixtos.
- Tiene una gran capacidad para manejar grandes conjuntos de datos con muchas características.

Desventajas:

- Requiere mas tiempo de entrenamiento que un solo árbol de decisión.
- Puede ser menos interpretativo, ya que el modelo final se construye a partir de muchos arboles.

2.5. Redes Neuronales

Las redes neuronales son un conjunto de algoritmos de aprendizaje automático (ML) diseñados para reconocer patrones. Funcionan de manera similar a la estructura de las neuronas del cerebro humano, lo que les permite identificar relaciones no lineales entre los datos de entrada y los resultados.

Datos que suelen usar:
Las redes neuronales son flexibles y pueden trabajar con diferentes tipos de datos. Son comunmente utilizadas para conjuntos de datos grandes, multidimensionales y

complejos. En el caso del Titanic, se pueden usar datos como la edad, sexo, clase de boleto, numero de hermanos, padres, e hijos a bordo, etc., para predecir la supervivencia.

Ventajas:

- Capacidad para capturar relaciones complejas y no lineales en los datos.
- Escalabilidad: Pueden manejar grandes volumenes de datos.
- Capacidad de Adaptación: Pueden ajustarse a nuevos datos, mejorando con el tiempo.

Desventajas:

- Requieren grandes cantidades de datos para entrenarse de manera eficiente.
- Interpretación limitada: Las redes neuronales suelen ser consideradas como "cajas negras", lo que dificulta la interpretación de como llegaron a una conclusión.
- Pueden ser propensas al sobreajuste (overfitting) si no se manejan correctamente.

2.6. Consideraciones para elegir un Modelo

Al elegir el modelo adecuado para resolver un problema de clasificación, como el reto del Titanic, se debe tener en cuenta varios factores:

- **Cantidad de datos:** Si se dispone de una gran cantidad de datos, una Red Neuronal puede ser más efectiva. Sin embargo, si los datos son limitados, Random Forest o la Regresión Logistica pueden ser mejores opciones.
- **Complejidad del problema:** Si las relaciones entre las variables son no lineales, un modelo como una Red Neuronal o rRandom Forest podría capturarlas mejor. Para relaciones más simples, la Regresión Logistica puede ser suficiente.
- **Interpretabilidad:** Si la interpretabilidad del modelo es importante (es decir, se necesita entender claramente por que se tomaron ciertas decisiones), la Regresión Logistica ofrece una mayor transparencia en comparacion con las Redes Neuronales y Random Forest.

- **Rendimiento:** Random Forest y Redes Neuronales tienden a tener un rendimiento mas robusto en problemas complejos. No obstante, este mayor rendimiento puede requerir mas recursos computacionales y un tiempo de entrenamiento más largo.

La elección del modelo dependerá de un equilibrio entre la cantidad de datos, la complejidad del problema, la necesidad de interpretación, y el rendimiento esperado.

3. Procesamiento de Datos

Para abordar el análisis de los datos del reto, se trabajó con dos conjuntos de datos proporcionados por Kaggle:

- **train.csv:** Este archivo contiene los datos utilizados para entrenar el modelo. Incluye características de los pasajeros, como su edad, sexo, clase de boleto, entre otros, así como la variable objetivo Survived, que indica si el pasajero sobrevivió o no al desastre.
- **test.csv:** Este archivo contiene las mismas características que el conjunto de entrenamiento, pero sin la variable objetivo. Fue utilizado para hacer las predicciones finales.

3.1. Limpieza de Datos

1. **Carga de los datos:** Los datos fueron cargados desde archivos CSV utilizando la biblioteca pandas. Se eliminó la columna PassengerId del conjunto de datos de entrenamiento, ya que esta variable no aporta información relevante para la predicción.
2. **Manejo de valores faltantes:** Los archivos train y test contenían valores faltantes en varias columnas clave, como Embarked, Age, Cabin y Fare. Para garantizar que el modelo pudiera procesar los datos correctamente, se aplicaron las siguientes estrategias para imputar los valores faltantes:
    - **Embarked:** En la columna Embarked, los valores faltantes fueron completados con el valor más común, que resultó ser 'S'. Este reemplazo se aplicó tanto al conjunto de entrenamiento como al de prueba.
    - **Age:** En la columna Age, se empleó la interpolación lineal, agrupando los pasajeros según su título (por ejemplo, Mr, Miss, Mrs, Master) para obtener una estimación más precisa de las edades faltantes en cada grupo.

- **Fare:** En la columna Fare, en este caso solo se tenía un valor faltante, este se rellenó con la media de las tarifas de los pasajeros que pertenecen a la misma clase (Pclass).
- **Cabin:** En la columna Cabin, la primera letra de cada valor representa el piso en el que estaba ubicada la cabina en el Titanic. Para simplificar el procesamiento, se extrajo esta letra como representación del valor de la columna.

  Dado que una gran cantidad de registros tenían valores faltantes en Cabin, estos se reemplazaron con 'N', indicando que no se tiene información sobre la cabina de esos pasajeros. Además, se identificó un valor único 'T', el cual no tenía un significado claro, por lo que se sustituyó por el valor más común, 'A'.

3. **Transformación de nuevas variables:**

   Durante el proceso de preprocesamiento, se generaron nuevas variables a partir de los datos originales, las variables añadidas fueron las siguientes:

   - **Title:** A partir de la columna Name, se extrajo el título de cada pasajero (por ejemplo, 'Mr', 'Mrs', 'Master', etc.). Posteriormente, se agruparon los títulos en categorías mas generales de la siguiente manera:
     - **Títulos de hombres:** Los títulos 'Mr', 'Don', 'Jonkheer','Capt','Rev' se agruparon en la categoría hombres.
     - **Títulos de mujeres:** Los títulos 'Lady','Miss','Mlle','Mme','Mrs','Ms','the Countess' se agruparon en la categoría mujeres.
     - **Títulos de rango alto:** Los títulos 'Col','Major','Sir','Dr' se agruparon en la categoría rango alto.
     - **Títulos de niños:** El título 'Master' no se agrupo con ni un otra variable y se dejó como estaba en los datos originales.
   - **Ticket:** La columna Ticket, que originalmente contenía una combinación de letras y numeros, fue procesada para extraer información útil. Se agregaron dos nuevas variables basadas en el número de ticket:
     - **Ticket_2letter:** Se extrajeron las primeras dos letras/numeros de cada valor de la columna Ticket, ya que las primeras dos letras nos aportan información valiosa, por ejemplo los tickets que comienzan con 'PC' mas del 60 % de estos pasajeros sobrevivieron al accidente.
     - **Ticket_len:** Se extrajo la longitud del ticket para cada pasajero y fue utilizada como una variable adicional.

- **Tipo_Fam:** Esta nueva variable fue creada para capturar el tamaño del grupo familiar de cada pasajero. Se calculó sumando las columnas SibSp (cantidad de hermanos/hermanas a bordo) y Parch (cantidad de padres/hijos a bordo), sumando 1 para incluir al propio pasajero. Posteriormente se agruparon las familias en Chico, Grande, Muy Grande, o si el pasajero viajaba solo.

4. **Codificación de variables categóricas:**

   Para que el modelo pudiera trabajar adecuadamente con las variables categoricas, estas fueron codificadas a valores numéricos:

   - **Sex:** La variable Sex, contenía valores de tipo texto ('male' o female'), esta fue convertida a valores numéricos. Se asignó el valor 0 a female y 1 a male.
   - **Embarked:** La variable Embarked que indicaba por donde los pasajeros abordaron el Titanic también fue transformada. Se asignaron valores numericos a cada puerto: S: 0, C: 1, Q:2.
   - **Cabin:** Como se mencionó anteriormente, la primera letra de cabina se extrajo y se asignaron valores numericos para cada letra: N: 0, A: 1, B: 2, C: 3, D: 4, E: 5, F: 6, G: 7.
   - **Title:** La variable Title, indicaba si la persona era hombre, mujer, niño o si pertenecía a un alto rango. Se asignaron valores númericos a cada uno de los titulos: 'Mr': 0 (hombres), 'Mrs': 1 (mujeres), 'Ra': 2 (Rango alto), 'Master': 3 (niños), 'Other': 4.
   - **Tipo Fam:** La variable Tipo Fam que indicaba el tamaño de la familia del pasajero. Se asignaron valores númericos a cada categoría de familia: 'Solo': 0, 'Chico': 1, 'Grande': 2, 'Muy Grande': 3.

   Finalmente, las variables categóricas fueron transformadas a variables dummy utilizando la función OneHotEncoder de sklearn. Esta técnica convierte cada categoria a una columna binaria, donde 1 indica la presencia de la categoría y un 0 su ausencia.

4. Análisis de Datos

Para el análisis de datos se usó el dataset limpio de **train.csv**. Se hizo una matriz de correlación con el fin de identificar las relaciones entre las variables, lo cual per-

mitió una mejor visualización de estas conexiones mediante gráficas. A continuación se muestra la matriz:

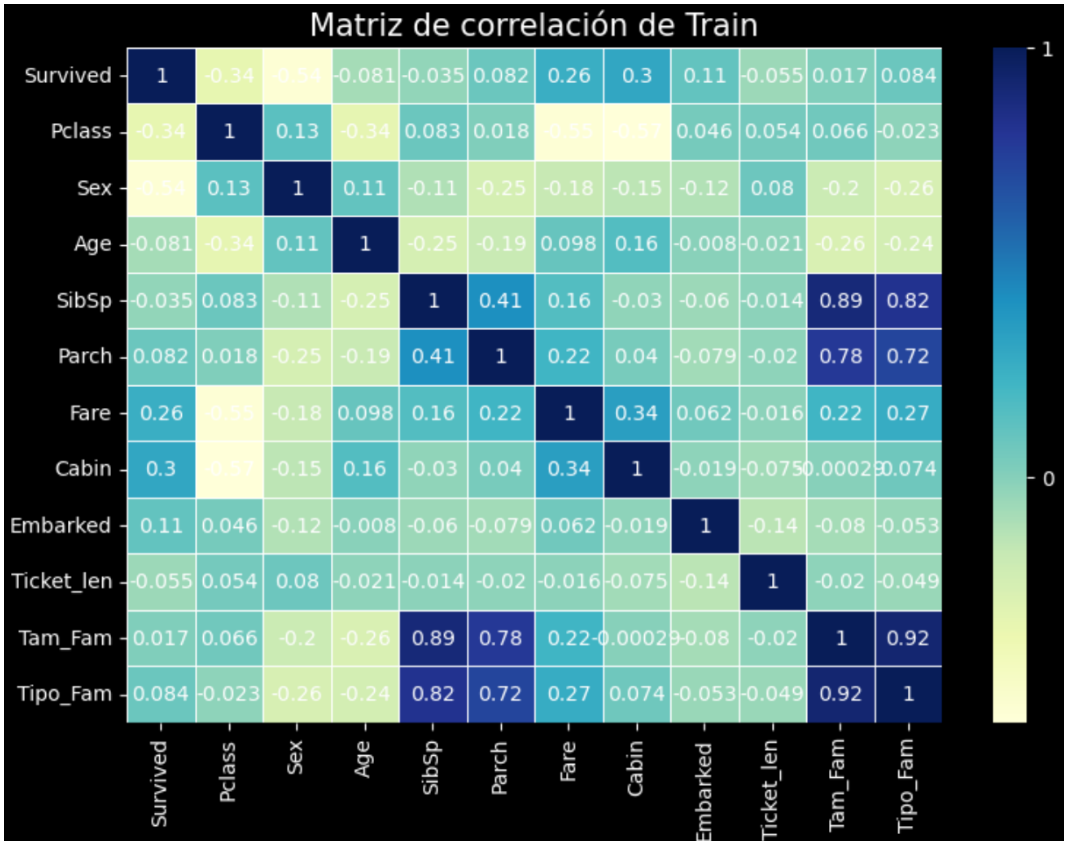

Figura 2: Matriz de correlación de **train.csv**

En esta se puede observar que la variable "Survived"tiene una correlación negativa con "Pclass" y "Sex", lo que indica que las personas que son de clase baja u hombres, tienen menos probabilidad de supervivencia. Por otro lado, en el caso de "Fare" y "Cabin", se observa que hay una correlación positiva, lo que sugiere que es más probable que los pasajeron que pagaron una tarifa más alta, vivan.

Igualmente, se puede notar que son justo estas variables las que más influyen en lo que es el propósito del problema: predecir la supervivencia de los pasajeros del Titanic. Ser mujer, pagar una tarifa alta, y por ende ser de clase alta, significa más probabilidad de supervivencia.

Habiendo dicho esto, se hicieron las siguientes gráficas:

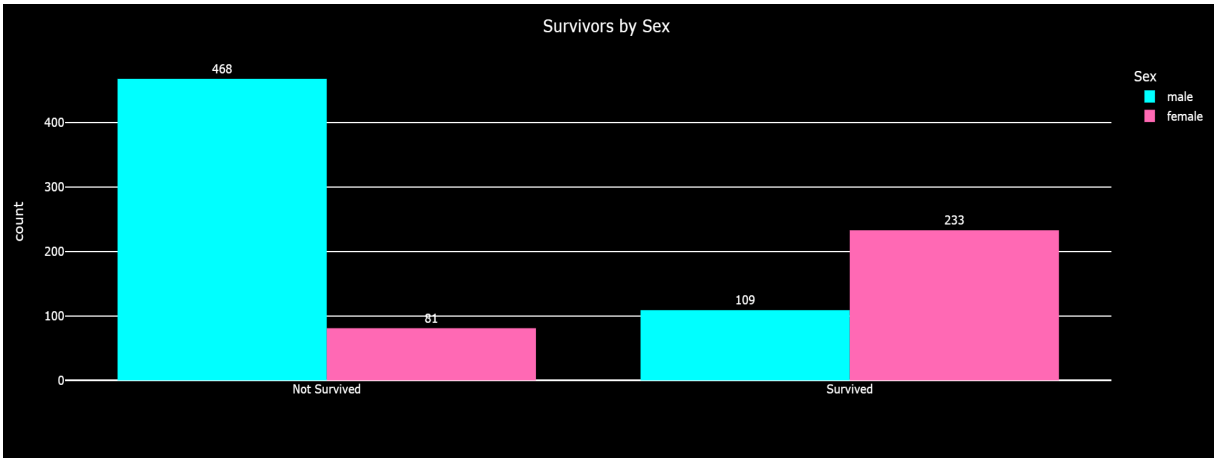

Figura 3: Gráfica de sobrevivientes por sexo

En la Figura 2 se puede observar la cantidad de personas que sobrevivieron o no, dependiendo de su sexo. En la categorías de personas que no sobrevivieron hay un total de 468 hombres y 81 mujeres. Por otro lado, en la categoría de sobrevivientes hay 109 hombres y 233 mujeres. Esto nos dice que tal y como se mencionó anteriormente, es más probable que un pasajero sobreviva si es mujer.

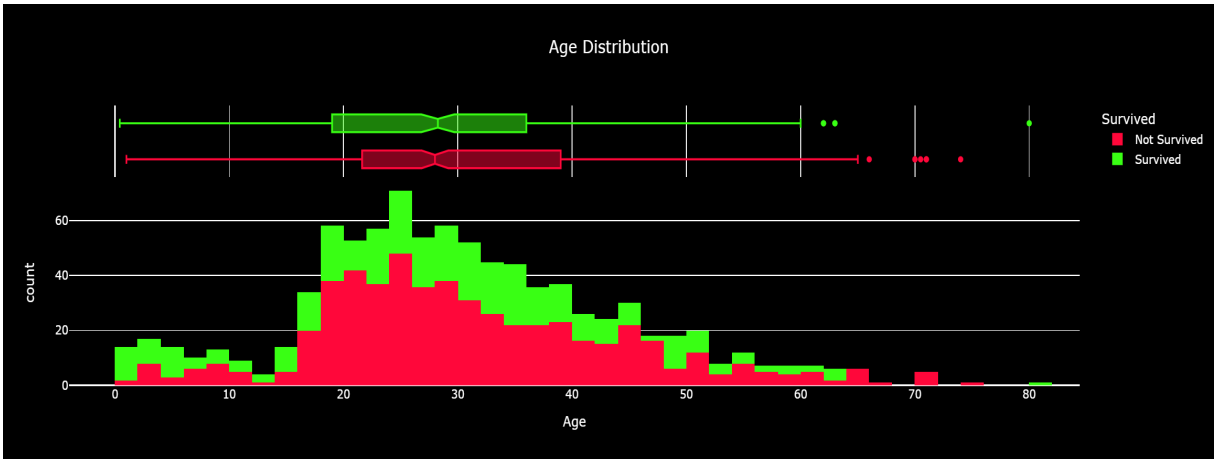

Figura 4: Gráfica de distribución de edades

La Figura 3 es una gráfica que muestra la distribución de edad de las personas que

sobrevivieron y las que no. En el boxplot verde, el cual corresponde a los sobrevivientes, se puede observar que la mayoría de los pasajeros tienen entre 20 y 40 años, con algunos valores atípicos. Asimismo, se observa cómo el rango intercuartílico no es tan amplio, lo que significa que las personas que no murieron tenían edades similares. Por otro lado, en el caso del boxplot rojo, el cual corresponde al de los no sobrevivientes, se puede ver que hay una mayor dispersión en las edades; sin embargo, el IQR se encuentra entre los 20 y 40 años.

Finalmente, se puede decir que observando los histogramas, se nota que a partir de los 50 años la cantidad de sobrevivientes disminuye, y que la mayoría es gente joven.

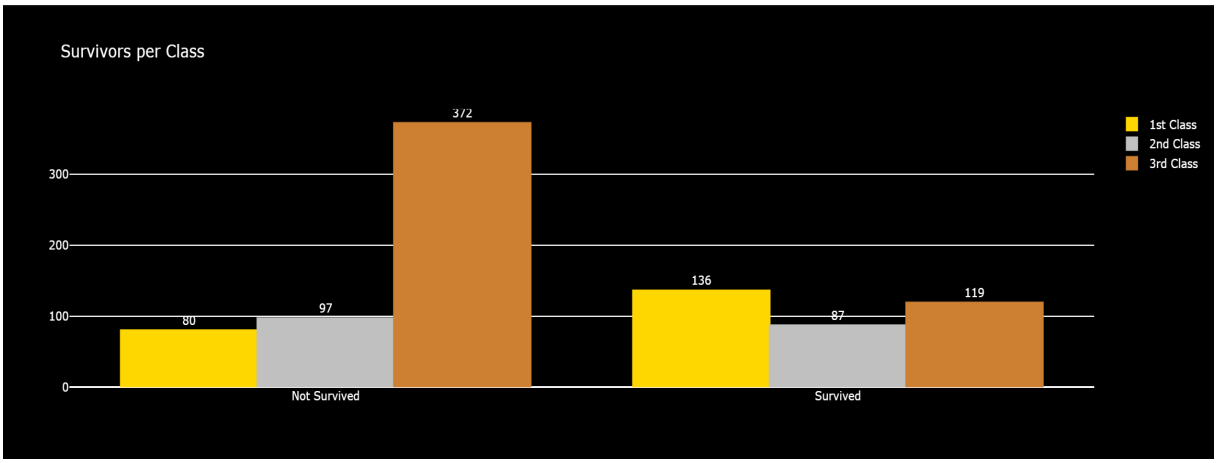

Figura 5: Gráfica de sobrevivientes por clase

La Figura 4 es un gráfico de barras el cual muestra la cantidad de sobrevivientes por clase. Tal y como se pensaba, las personas con menos posibilidad de supervivencia son aquellas que estaban en la tercera clase, con 372 pasajeros. Contrario a esto, los pasajeros en primera clase, a pesar de ser pocos, tienen una mayor presencia en las personas que sobrevivieron.

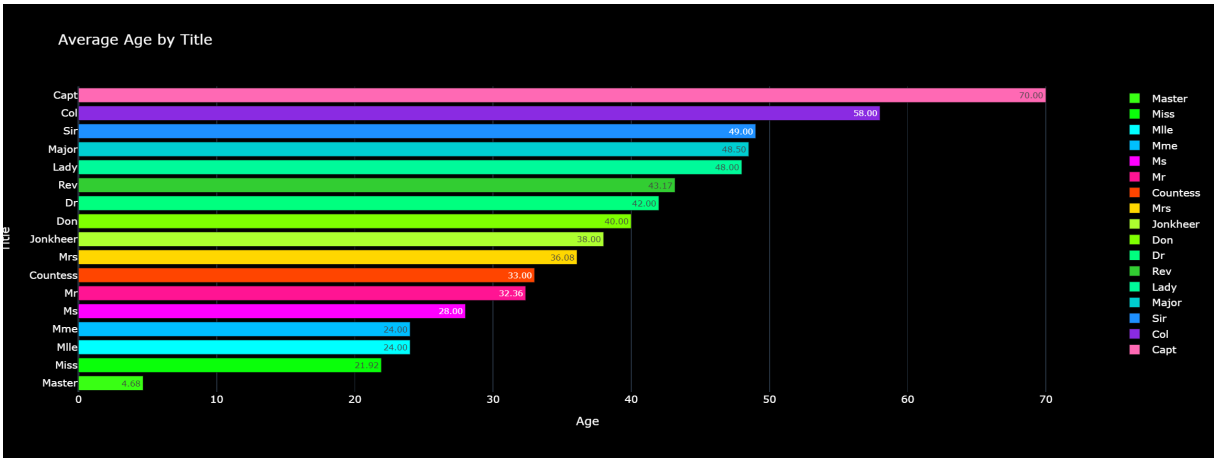

Figura 6: Gráfica de edad media por título

Finalmente, en esta última gráfica de la parte del análisis de datos se puede observar la edad media por título. Para poder hacer esto, se tuvo que sacar la cantidad de personas por título. La manera en la que se distribuyen en la siguiente:

- Mr: 517
- Miss: 182
- Mrs: 125
- Master: 40
- Capitán: 1
- Extra: 26

Habiendo dicho esto, la edad del capitán del barco era 70 años, la media de la edad de los hombres con el título “Mr” es 32.36 años, la de las mujeres con el título “Miss” es 21.92 y con el título “Mrs” es 36.08, y la de los niños con el título “Master” es 4.68. Existen otros títulos los cuales cuentan con muy pocas personas, es por eso que se agrupan por aparte.

5. Modelos Predictivos

Los tres modelos seleccionados fueron entrenados utilizando el mismo conjunto de variables predictoras. Para el procesamiento de las variables categóricas como se mencionó anteriormente, se empleó la técnica de One Hot Encoding, lo que permitió convertir dichas variables en representaciones binarias, facilitando su uso por parte de los algoritmos.

Las variables utilizadas para entrenar el modelo fueron:

```
# Variable objetivo o a predecir
y = df_train_cleaned['Survived']

# Caracteristicas seleccionadas
features = ['Pclass','Title','Embarked','Tipo_Fam','Ticket_len','
    Ticket_2letter']

# Seleccion de las caracteristicas
X = df_train_cleaned[features]
```

Tras entrenar los modelos, se realizaron predicciones en el conjunto de datos de prueba. Las predicciones fueron guardadas en un archivo csv para poder entregarlas en la competencia de Kaggle, pero también se puede revisar la calificación de dicha competencia en los notebooks de cada modelo.

5.1. Regresión Logística

```
# Dividimos los datos en entrenamiento y validacion (70%
    entrenamiento, 30% validacion)
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size
    =0.3, random_state=42)

# Procesador para transformar las variables

# Aqui agregamos un SimpleImputer para valores numericos (si es
    necesario) y con OneHotEncoder para categoricas
preprocessor = ColumnTransformer(transformers=[
    ('onehot', OneHotEncoder(handle_unknown='ignore'), features)
])

# Definimos el modelo de regresion logistica con los hiperparametros
    proporcionados
modelo = Pipeline(steps=[
```

```
    ('preprocessor', preprocessor),
    ('model', LogisticRegression(
        random_state=42,
        max_iter=1000,
        C=100,                          # Hiperparametro C ajustado
        class_weight=None,              # Sin balanceo de clases
        penalty='l2',                   # Penalizacion L1
        solver='newton-cg'))])

# Entrenamiento del modelo
modelo.fit(X_train, y_train)

# Predicciones en el conjunto de validacion
y_pred = modelo.predict(X_val)
```

El modelo de Regresión Logística fue implementado utilizando un pipeline con la biblioteca sklearn, donde se incorporó el preprocesamiento de datos mediante One Hot Encoding y la estandarización de variables numéricas. Para la etapa de modelado, se utilizó la regularización L2, con un valor ajustado del hiperparámetro C igual a 100, lo que permitió controlar el grado de regularización. El algoritmo fue optimizado con el solver 'newton-cg'y se configuró con un máximo de 1000 iteraciones. Esta configuración fue seleccionada para mejorar la precisión sin caer en sobreajuste, aprovechando la robustez de la regresión logstica en tareas de clasificación.

5.2. Random Forest

```
# Dividimos los datos en entrenamiento y validacion (70%
    entrenamiento, 30% validacion)
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size =
    0.3, random_state = 42)

# Procesador para transformar a variables dummy
preprocessor = ColumnTransformer(transformers =[('onehot',
    OneHotEncoder(handle_unknown = "ignore"), features), ])

# Definimos el modelo y hacemos la transformacion a variables dummy
modelo = Pipeline(steps=[('preprocessor', preprocessor),
    ('model', RandomForestClassifier(random_state = 42, n_estimators
        = 500, max_depth = 5))])

# Entrenamiento del modelo
modelo.fit(X_train, y_train)

# Predicciones en el conjunto de validacion
```

```
y_pred = modelo.predict(X_val)
```

El modelo de Random Forest fue implementado con la biblioteca sklearn. Se utilizó una cantidad de 500 estimadores (arboles de decisión), con un valor maximo de profundiad de 5 de esta forma evitando el sobreajuste. Este algoritmo fue seleccionado debido a su capacidad para manejar tanto variables categóricas como numéricas de manera eficiente.

5.3. Red Neuronal

```
# Dividimos los datos en entrenamiento y validacion (70%
    entrenamiento, 30% validacion)

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size =
    0.3, random_state = 42)

# Procesador para transformar a variables dummy y escalar
    caracteristicas numericas
preprocessor = ColumnTransformer(transformers=[
    ('onehot', OneHotEncoder(handle_unknown='ignore'), ['Pclass','
       Title','Embarked','Tipo_Fam','Ticket_2letter']),
    ('scaler', StandardScaler(), ['Ticket_len'])])

# Definimos el modelo utilizando una red neuronal (MLPClassifier)
    con la funcion de activacion 'tanh'
modelo = Pipeline(steps = [
    ('preprocessor', preprocessor),
    ('model', MLPClassifier(random_state=42,
                            max_iter=1000,  # Incrementamos el
                               numero de iteraciones
                            hidden_layer_sizes=(350, 150, 100),  #
                               Aumentamos el tamano de las capas
                            activation='tanh',
                            alpha=0.01,  # Regularizacion
                            learning_rate_init=0.01))])

# Entrenamiento del modelo
modelo.fit(X_train, y_train)

# Predicciones en el conjunto de validacion
y_pred = modelo.predict(X_val)
```

El modelo de Red Neuronal fue implementado mediante un pipeline en la biblioteca sklearn. El modelo fue entrenado utilizando un MLPClassifier con tres capas

ocultas de tamaños 350, 150 y 100 neuronas, respectivamente, lo que permitió aumentar la capacidad de aprendizaje del modelo. Se utilizó la función de activación 'tanh' y una regularización con el hiperparámetro alpha ajustado a 0.01, para prevenir el sobreajuste. Además, el modelo fue configurado con una tasa de aprendizaje inicial de 0.01 y un máximo de 1000 iteraciones para asegurar una convergencia adecuada.

6. Resultados

Para evaluar el rendimiento de cada modelo, se subieron los archivos de predicciones a la plataforma de Kaggle, donde se compararon los resultados con las etiquetas reales del conjunto de prueba. La manera en que funciona es que Kaggle calculó el número de predicciones correctas de cada modelo, generando un porcentaje de exactitud. Los resultados son los siguientes:

6.1. Regresión Logística

	Precision	**Recall**	**F1-Score**	**Support**
Class 0	0.83	0.91	0.87	175
Class 1	0.80	0.66	0.72	93
Accuracy			0.82	268
Macro Avg	0.82	0.79	0.80	268
Weighted Avg	0.82	0.82	0.82	268

Cuadro 1: Informe de clasificación con puntuación de precisión de 0.8246

6.1.1. Interpretación de Resultados

- **Clase 0 o que No sobrevivió**:
  - **Precision:** 0.83. De todos los pasajeros que el modelo predijo como que no sobrevivieron, el 83 % efectivamente no sobrevivió.
  - **Recall:** 0.91. De todos los pasajeros que realmente no sobrevivieron, el modelo identificó correctamente al 91 %.
  - **F1-Score:** 0.87. Es el promedio entre la precisión y el recall, lo que indica un buen balance entre ambos para la clase de "no sobrevivió".

- **Support:** 175. Este valor indica que hubo 175 pasajeros que efectivamente no sobrevivieron, es decir, los casos reales de la clase 0 en los datos de prueba.

- **Clase 1 o que Sobrevivió**:

  - **Precision:** 0.80. De todos los pasajeros que el modelo predijo como que sobrevivieron, el 80 % realmente sobrevivió.
  - **Recall:** 0.66. De todos los pasajeros que realmente sobrevivieron, el modelo identificó correctamente al 66 %.
  - **F1-Score:** 0.72. Este valor es menor que el de la clase 0 debido a la dificultad del modelo para identificar correctamente a todos los sobrevivientes.
  - **Support:** 93. Este valor indica que 93 pasajeros efectivamente sobrevivieron, es decir, los casos reales de la clase 1 en los datos de prueba.

El modelo tiene un rendimiento en general aceptable, con una exactitud global del 0.8246, lo que nos dice que predice correctamente en el 82 % de los casos. En cuanto a las métricas globales, el promedio macro, que da el mismo peso a cada clase sin importar su tamaño, nos da una precisión, recall y F1-Score de 0.82, 0.79 y 0.80, respectivamente. Por otro lado, el promedio ponderado, que tiene en cuenta el desequilibrio entre las clases, también reporta una precisión, recall y F1-Score de 0.82.

Sin embargo, aunque el modelo funciona bien en general, tiene dificultades para identificar correctamente a los sobrevivientes o a la clase 1, ya que su recall es de solo 0.66, lo que significa que no logra identificar a muchos sobrevivientes reales. A pesar de que predice con mayor precisión a los pasajeros que no sobrevivieron, creemos que el modelo podría ser mejor con un par de ajustes adicionales, tal vez algún tipo de técnica de balanceo de clases o remuestreo, y así poder mejorar su capacidad de identificar a los sobrevivientes.

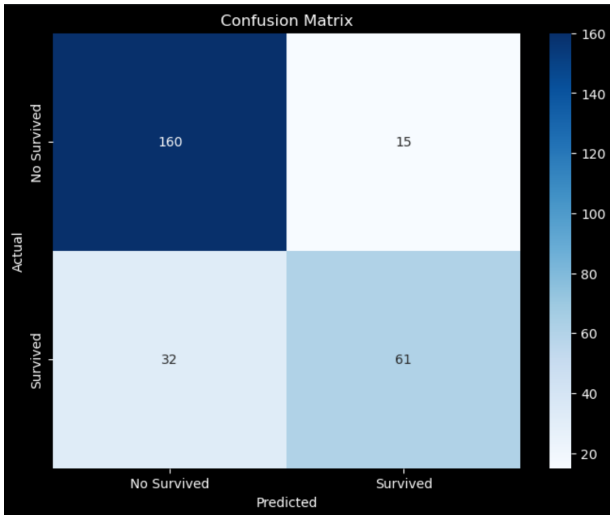

Figura 7: Matriz de confusión del modelo de Regresión Logística

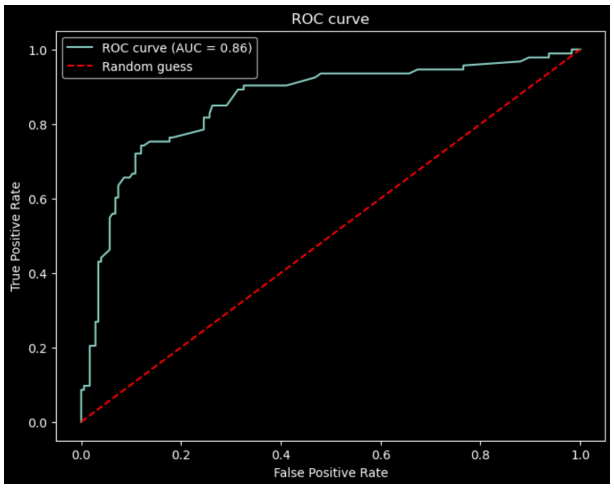

Figura 8: Curva ROC del modelo de Regresión Logística

- AUC (Area Under the Curve): El AUC es 0.86. Esto nos dice que el modelo tiene un buen rendimiento al diferenciar entre clases. Un AUC de 1.0 significa un clasificador perfecto, mientras que un AUC de 0.5 indica un rendimiento equivalente a un modelo que adivina aleatoriamente (como una monea que lanzamos al aire).

6.2. Random Forest

6.2.1. Interpretación de Resultados

	Precision	Recall	F1-Score	Support
Class 0	0.83	0.94	0.88	175
Class 1	0.85	0.65	0.73	93
Accuracy			0.84	268
Macro Avg	0.84	0.79	0.81	268
Weighted Avg	0.84	0.84	0.83	268

Cuadro 2: Informe de clasificación con puntuación de precisión de 0.8358

- **Clase 0 o que No sobrevivió**:
  - **Precision:** 0.83. De todos los pasajeros que el modelo predijo como que no sobrevivieron, el 83 % efectivamente no sobrevivió.
  - **Recall:** 0.94. De todos los pasajeros que realmente no sobrevivieron, el modelo identificó correctamente al 94 %.
  - **F1-Score:** 0.88. Es el promedio entre la precisión y el recall, lo que indica un buen balance entre ambos para la clase de "no sobrevivió".
  - **Support:** 175. Este valor indica que hubo 175 pasajeros que efectivamente no sobrevivieron, es decir, los casos reales de la clase 0 en los datos de prueba.
- **Clase 1 o que Sobrevivió**:
  - **Precision:** 0.85. De todos los pasajeros que el modelo predijo como que sobrevivieron, el 85 % realmente sobrevivió.
  - **Recall:** 0.65. De todos los pasajeros que realmente sobrevivieron, el modelo identificó correctamente al 65 %.
  - **F1-Score:** 0.73. Este valor es menor que el de la clase 0 debido a la dificultad del modelo para identificar correctamente a todos los sobrevivientes.
  - **Support:** 93. Este valor indica que 93 pasajeros efectivamente sobrevivieron, es decir, los casos reales de la clase 1 en los datos de prueba.

El modelo presenta un rendimiento en general bueno, con una exactitud global del 0.84, lo que nos dice que predice correctamente en el 84 % de los casos. En cuanto a las métricas globales, el promedio macro, que asigna el mismo peso a cada clase sin importar su tamaño, reporta una precisión, recall y F1-Score de 0.84, 0.79 y 0.81, respectivamente. Por otro lado, el promedio ponderado, que toma en cuenta el desequilibrio entre las clases, refleja una precisión, recall y F1-Score de 0.84, 0.84 y 0.83.

Sin embargo, aunque el modelo funciona bien en general, presenta dificultades para identificar correctamente a los sobrevivientes o a la clase 1, dado que su recall es de solo 0.65, lo cual es peor que el modelo de Regresión Logística., lo que significa que no logra identificar a una cantidad significativa de los sobrevivientes reales. A pesar de que predice con mayor precisión a los pasajeros que no sobrevivieron, creemos que el modelo podría mejorar mediante ajustes adicionales, como la aplicación de alguna técnica de balanceo de clases o remuestreo, lo que podría aumentar su capacidad para identificar mejor a los sobrevivientes.

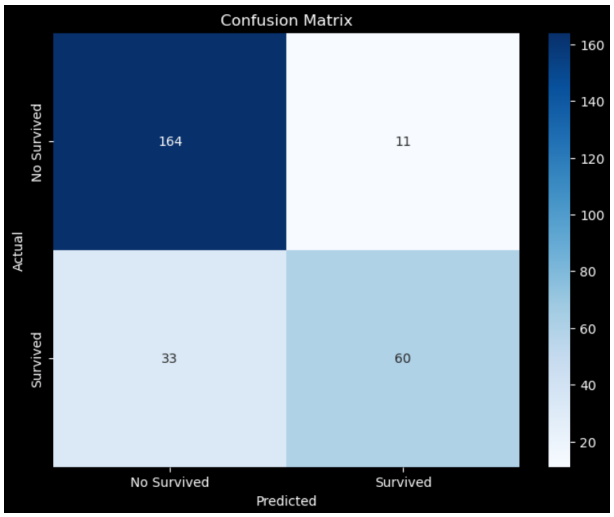

Figura 9: Matriz de confusión del modelo de Random Forest

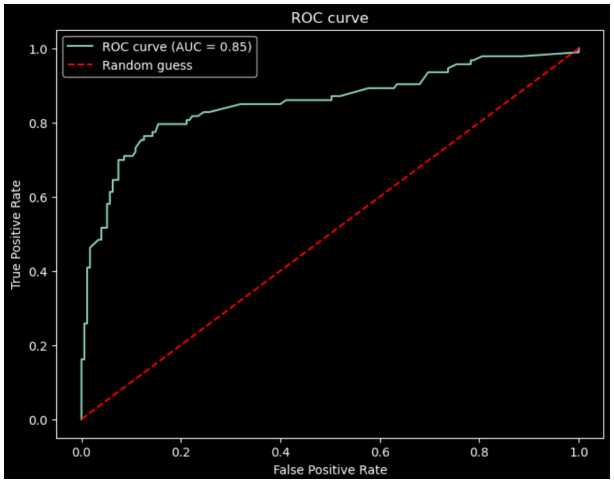

Figura 10: Curva ROC del modelo de Random Forest

- AUC (Area Under the Curve): El AUC es 0.85. Esto nos dice que el modelo tiene un buen rendimiento al diferenciar entre clases. Un AUC de 1.0 significa un clasificador perfecto, mientras que un AUC de 0.5 indica un rendimiento equivalente a un modelo que adivina aleatoriamente (como una monea que lanzamos al aire).

6.3. Red Neuronal

6.3.1. Interpretación de Resultados

	Precision	**Recall**	**F1-Score**	**Support**
Class 0	0.86	0.90	0.88	175
Class 1	0.80	0.72	0.76	93
Accuracy			0.84	268
Macro Avg	0.83	0.81	0.82	268
Weighted Avg	0.84	0.84	0.84	268

Cuadro 3: Informe de clasificación con puntuación de precisión de 0.8396

- **Clase 0 o que No sobrevivió**:
  - **Precision:** 0.86. De todos los pasajeros que el modelo predijo como que no sobrevivieron, el 86 % efectivamente no sobrevivió.

- **Recall:** 0.90. De todos los pasajeros que realmente no sobrevivieron, el modelo identificó correctamente al 90 %.
- **F1-Score:** 0.88. Es el promedio entre la precisión y el recall, lo que indica un buen balance entre ambos para la clase de "no sobrevivió".
- **Support:** 175. Este valor indica que hubo 175 pasajeros que efectivamente no sobrevivieron, es decir, los casos reales de la clase 0 en los datos de prueba.

- **Clase 1 o que Sobrevivió**:
  - **Precision:** 0.80. De todos los pasajeros que el modelo predijo como que sobrevivieron, el 80 % realmente sobrevivió.
  - **Recall:** 0.72. De todos los pasajeros que realmente sobrevivieron, el modelo identificó correctamente al 72 %.
  - **F1-Score:** 0.76. Este valor es menor que el de la clase 0 debido a la dificultad del modelo para identificar correctamente a todos los sobrevivientes, cabe destacar que es mejor que los dos modelos anteriores.
  - **Support:** 93. Este valor indica que 93 pasajeros efectivamente sobrevivieron, es decir, los casos reales de la clase 1 en los datos de prueba.

La Red Neuronal presenta un rendimiento en general mejorado, con una exactitud global del 0.84, lo que nos dice que predice correctamente en el 84 % de los casos. En cuanto a las métricas globales, el promedio macro, que asigna el mismo peso a cada clase sin importar su tamaño, reporta una precisión, recall y F1-Score de 0.83, 0.81 y 0.82, respectivamente. Por otro lado, el promedio ponderado, que toma en cuenta el desequilibrio entre las clases, refleja una precisión, recall y F1-Score de 0.84 para todos.

Aunque el modelo funciona bien mejor en comparación con los modelos pasados, sigue presentando dificultades para identificar correctamente a los sobrevivientes o a la clase 1, dado que su recall es de solo 0.72, lo cual es mejor que los modelos ya vistos, esto presenta una mejora sustancial al momento de poder identificar correctamente a los de la clase 1, pero aun así significa que no logra identificar a una cantidad significativa de los sobrevivientes reales, en comparación con la clase 0. A pesar de que predice con mayor precisión a los pasajeros que no sobrevivieron, creemos que la red neuronal podría mejorar mediante ajustes adicionales, como la aplicación de alguna técnica de balanceo de clases o remuestreo, lo que podría aumentar su capacidad para identificar mejor a los sobrevivientes.

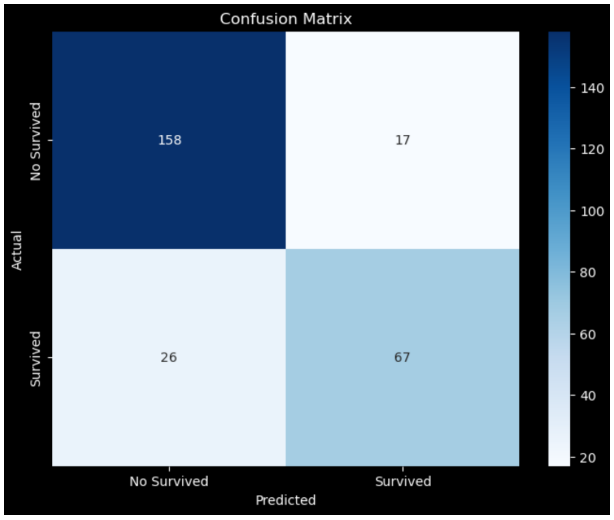

Figura 11: Matriz de confusión del modelo de Redes Neuronales

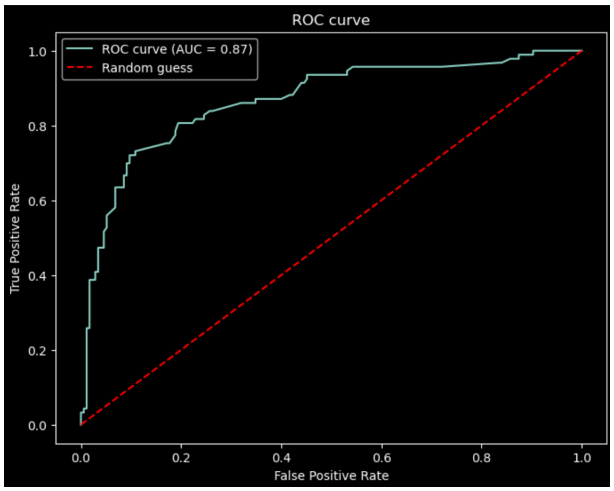

Figura 12: Curva ROC del modelo de Redes Neuronales

- AUC (Area Under the Curve): El AUC es 0.87. Esto nos dice que el modelo tiene un buen rendimiento al diferenciar entre clases. Un AUC de 1.0 significa un clasificador perfecto, mientras que un AUC de 0.5 indica un rendimiento equivalente a un modelo que adivina aleatoriamente (como una monea que lanzamos al aire).

6.4. Kaggle Submissions

En esta sección se muestran los resultados obtenidos de las distintas submissions realizadas en la plataforma de Kaggle para el reto del Titanic. Cada submission corresponde a uno de los modelos evaluados: Regresión Logística, Random Forest y Red Neuronal.

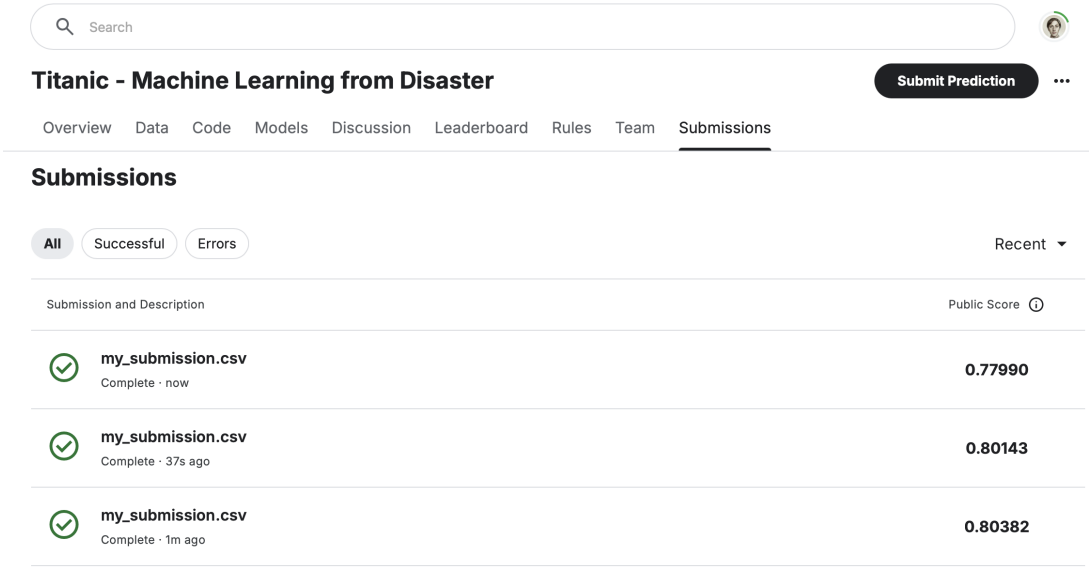

Figura 13: Puntuaciones obtenidas en las submissions en Kaggle

Como se puede observar, los resultados de los modelos coinciden con las precisiones obtenidas en los conjuntos de prueba, siendo las redes neuronales y Random Forest los modelos con mejor desempeño.

6.5. Comparación de Resultados

Modelo	**Precisión (%)**
Regresión Logística	$\frac{326}{418} = 77{,}99\,\%$
Random Forest	$\frac{335}{418} = 80{,}14\,\%$
Red Neuronal	$\frac{336}{418} = 80{,}38\,\%$

Cuadro 4: Resultados de Precisión para Diferentes Modelos

La tabla nos dice que, en términos de precisión, los modelos más avanzados como Random Forest y las Redes Neuronales superan a la Regresión Logística, aunque las diferencias no son tan grandes.

Esto nos dice que los datos utilizados no son suficientemente complejos o que ya se ha extraído gran parte de la información útil en los modelos más simples, como descrito en la interpretación de cada modelo, se recomienda hacer transformaciones más robustas a los datos, que ayuden a que cada modelo pueda manejarlos de forma que sea beneficioso para el ajuste de estos ayudando a la precisión del modelo en general.

7. Conclusión

A lo largo de este proyecto, se analizaron múltiples enfoques de modelado predictivo para estimar la probabilidad de supervivencia de los pasajeros del Titanic, utilizando técnicas como Regresión Logística, Random Forest, y Redes Neuronales.

Cada modelo presentó ventajas y desventajas, demostrando que la elección del metodo depende tanto de la naturaleza de los datos como del objetivo específico del análisis. Los resultados obtenidos reflejan que, si bien la Regresión Logística ofrece una interpretación clara de las relaciones entre las variables y es eficaz en escenarios donde los datos son linealmente separables, los modelos de Random Forest y Redes Neuronales lograron mejorar la precisión debido a su capacidad de interacciones no lineales y patrones complejos en los datos. No obstante, estos modelos más avanzados requieren una mayor potencia computacional y pueden resultar menos interpretables que la Regresión Logística. Sin embargo, Random Forest, gracias a su robustez y manejo de variables no lineales, y las Redes Neuronales, por su flexibilidad capacidad de ajustarse a los datos, mostraron un rendimiento superior en términos de precisión.

Finalmente, este análisis no solo permitió mejorar el modelo predictivo, sino que tambien resaltó la importancia de un adecuado procesamiento de datos, y la consideración de diversas estrategias de modelado. En futuros proyectos, sería interesante explorar técnicas adicionales de optimización de hiperparámetros y enfoques como el ensamblado de modelos para mejorar aún mas los resultados. En conclusión, aunque cada técnica tiene su lugar en el análisis predictivo, las metodologías avanzadas como Random Forest y Redes Neuronales demostraron ser más eficaces en este caso, constribuyendo a una predicción más precisa, salvando vidas.

8. Referencias

- IBM Cloud Education. (2020, 31 de julio). What are neural networks? IBM. Recuperado de https://www.ibm.com/cloud/learn/neural-networks
- Encyclopedia Titanica. (n.d.). Decks and cabins of the RMS Titanic. Encyclopedia Titanica. Recuperado de https://www.encyclopedia-titanica.org/titanic-deckplans.html
- Scikit-learn. (n.d.). Logistic regression. Scikit-learn. Recuperado de https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
- Scikit-learn. (n.d.). Random forest. Scikit-learn. Recuperado de https://scikit-learn.org/stable/modules/ensemble.html#random-forests

A continuación se muestran enlaces a notebooks de Google Colab que documentan el procesamiento, análisis y modelado de datos.

1. **Procesamiento de Datos** - https://colab.research.google.com/drive/1Twjp_jHX9838dhAf-YXDy9vxVkJJQGd3?usp=sharing
2. **Análisis de Datos** - https://colab.research.google.com/drive/1r9q_cZiGEdMJdRWi-ccKK4OqGY4OoEhM?usp=sharing
3. **Regresión Logística** - https://colab.research.google.com/drive/1MM-mUSadCH7OhYzrtXABON51o5UX__z-?usp=sharing
4. **Random Forest** - https://colab.research.google.com/drive/1IQtlA5x__OPkIfqYaqdw44kr050kVJIa?usp=sharing
5. **Red Neuronal** - https://colab.research.google.com/drive/1gtv5DIy41GF_HO3ujLVS2_6bkETrtNRG?usp=sharing

Repositorio de GitHub: https://github.com/Lautaro000/Inteligencia-Artificial-Avanzada-para-la-Ciencia-de-Datos-I