# Proyecto integrador: Sobreviviendo del Titanic

El RMS Titanic zarpó en su viaje inaugural en 1912, cruzando el Atlántico desde Southampton, Inglaterra, hasta la ciudad de Nueva York. El barco nunca completó el viaje, hundiéndose hasta el fondo del océano Atlántico después de chocar con un iceberg, derribando a 1.502 de los 2.224 pasajeros a bordo. En este proyecto, creará un modelo de regresión logística que prediga qué pasajeros sobrevivieron al hundimiento del Titanic, en función de características como la edad y la clase. Kaggle proporciona los datos que usaremos para entrenar nuestro modelo. ¡Siéntete libre de mejorar el modelo por tu cuenta y enviarlo a la competencia Kaggle Titanic!

Adjunto a este documento encontrarás el archivo `script.py`, que te servirá de guía para completar paso a paso nuestro proyecto integrador.

## Cargar los datos

1. En una nueva libreta, en la plataforma Kaggle, importa los datos de la competencia 'Titanic - Machine Learning from Disaster'. A partit del archivo 'train.csv', crea un marco de datos 'passengers'. Imprime la cabecera de este marco e inspecciona las columnas. ¿Qué características serías más útiles para predecir la supervivencia de un pasajero?

## Limpiar los datos

2. Dado el dicho, "las mujeres y los niños primero", el sexo y la edad parecen ser buenas características para predecir la supervivencia. Mapeemos los valores de texto en la columna `Sex` a un valor numérico. Actualice `Sex` de modo que todos los valores femeninos se reemplacen con 1 y todos los valores masculinos se reemplacen con 0.
3. Echemos un vistazo a la edad. Imprima `passengers['Age'].values`. Puede ver que tenemos varios valores faltantes, o `nan`s. Rellene todos los valores de edad vacíos en pasajeros con la edad media.
4. Dado el estricto sistema de clases a bordo del Titanic, utilicemos la columna `Pclass`, o la clase de pasajeros, como otra característica. Cree una nueva columna llamada `FirstClass` que almacene 1 para todos los pasajeros en primera clase y 0 para todos los demás pasajeros.
5. Cree una nueva columna llamada `SecondClass` que almacene 1 para todos los pasajeros en segunda clase y 0 para todos los demás pasajeros. Imprima pasajeros e inspeccione el DataFrame para asegurarse de que se hayan realizado todas las actualizaciones.

# Seleccionar y dividir los datos

6. Ahora que hemos limpiado nuestros datos, seleccionemos las columnas sobre las que queremos construir nuestro modelo. Seleccione las columnas `Sex`, `Age`, `FirstClass` y `SecondClass` y guárdelas en una variable denominada `features`. Seleccione la columna `Survived` y guárdela en una variable llamada `survival`.
7. Divida los datos en conjuntos de entrenamiento y prueba utilizando el método `train_test_split()` de `sklearn`. Usaremos el conjunto de entrenamiento para entrenar el modelo y el conjunto de prueba para evaluar el modelo.
8. Dado que la implementación de la regresión logística de `sklearn` utiliza la regularización, necesitamos escalar nuestros datos de características. Cree un objeto `StandardScaler`, y emplee `.fit_transform()` en las características de entrenamiento y `.transform()` en las características de prueba.
9. Cree un modelo `LogisticRegression` con `sklearn` y `.fit()` en los datos de entrenamiento. El ajuste del modelo realizará un descenso de gradiente para encontrar los coeficientes de características que minimicen la pérdida logarítmica para los datos de entrenamiento.
10. Emplee el método `.score()` en el modelo en los datos de entrenamiento e imprime el puntaje de entrenamiento. La puntuación del modelo en los datos de entrenamiento ejecutará los datos a través del modelo y hará clasificaciones finales de supervivencia para cada pasajero en el conjunto de entrenamiento. La puntuación devuelta es el porcentaje de clasificaciones correctas o la precisión.
11. Emplee el método `.score()` en el modelo en los datos de la prueba e imprime la puntuación de la prueba. De manera similar, calificar el modelo en los datos de prueba ejecutará los datos a través del modelo y hará clasificaciones finales sobre supervivencia para cada pasajero en el conjunto de prueba. ¿Qué tan bien se desempeñó su modelo?
12. Imprime los coeficientes de característica determinados por el modelo. ¿Qué característica es más importante para predecir la supervivencia en el hundimiento del Titanic?

# Predicciones con el modelo

13. Usemos nuestro modelo para hacer predicciones sobre la supervivencia de algunos fatídicos pasajeros. En el editor de código se proporciona información para el pasajero de 3.ª clase `Jack` y la pasajera de 1.ª clase `Rose`, almacenada en matrices NumPy. Las matrices almacenan 4 valores de característica, en el siguiente orden:
    1. `Sex`, representado por un 0 para hombre y un 1 para mujer
    2. `Age`, representada como un número entero en años
    3. `FirstClass`, con un 1 que indica que el pasajero está en primera clase

    4. `SecondClass`, con un 1 que indica que el pasajero está en segunda clase
    5. Una tercera matriz, `You`, también se proporciona en el editor de código con valores de característica vacíos. Descomente la línea que lo contiene y actualice la matriz con su información o la información de algún pasajero ficticio. ¡Asegúrese de ingresar todos los valores como flotantes con un .!
14. Combine `Jack`, `Rose`, and `You` into a single `NumPy` array named `sample_passengers`.
15. Dado que nuestro modelo de regresión logística se entrenó en datos de características escaladas, también debemos escalar los datos de características sobre los que estamos haciendo predicciones. Con el objeto `StandardScaler` creado anteriormente, aplique su método `.transform()` a `sample_passengers` y guarde el resultado en `sample_passengers`. Imprima `sample_passengers` para ver las características escaladas.
16. ¿Quién sobrevivirá y quién se hundirá? Use el método `.predict()` de su modelo en `sample_passengers` e imprima el resultado para averiguarlo.

    ¿Quiere ver las probabilidades que llevaron a estas predicciones? Llame al método `.predict_proba()` de su modelo en `sample_passengers` e imprima el resultado. La primera columna es la probabilidad de que un pasajero muera en el Titanic, y la segunda columna es la probabilidad de que un pasajero sobreviva al hundimiento (que fue calculado por nuestro modelo para tomar la decisión de clasificación final).