In [1]:
```
#importação de bibliotecas
from sklearn.model_selection import train_test_split
from sklearn.preprocessing   import StandardScaler
from sklearn.metrics         import classification_report
from tensorflow              import keras
import tensorflow as tf
import pandas     as pd
import numpy      as np

import os
```

```
2025-07-28 01:53:38.680941: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register cuFFT factory: At
tempting to register factory for plugin cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1753667618.966790      13 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for pl
ugin cuDNN when one has already been registered
E0000 00:00:1753667619.044220      13 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for
plugin cuBLAS when one has already been registered
```

In [2]:
```
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/titanic/train.csv
/kaggle/input/titanic/test.csv
/kaggle/input/titanic/gender_submission.csv
```

In [3]:
```
csv_train_path  = '/kaggle/input/titanic/train.csv'

raw_df    = pd.read_csv(csv_train_path)
```

In [4]:
```
target_column       = ['Survived']
training_columns    = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']
categorical_columns = ['Sex', 'Pclass', 'Embarked']
```

In [5]:
```
#treat data
def treat_data(raw_df, training_columns, target_column, trainning = True):
    all_columns = training_columns + target_column if trainning else training_columns
```

```
    treated_df = raw_df.copy()
    treated_df['Age'] = treated_df['Age'].fillna(treated_df['Age'].mean())
    treated_df['Fare'] = treated_df['Fare'].fillna(treated_df['Fare'].mean())

    treated_df = treated_df[all_columns]
    treated_df = treated_df.dropna()

    treated_df = pd.get_dummies(treated_df, columns=categorical_columns, drop_first=True, dtype=int)

    return treated_df

treated_df = treat_data(raw_df, training_columns, target_column)
expanded_training_columns = list(treated_df.columns)
expanded_training_columns.remove(target_column[0])
```

In [6]:
```
X = treated_df[expanded_training_columns]
y = treated_df[target_column]

X_train = X.values
y_train = y.values

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

In [7]:
```
model = keras.Sequential([
    keras.Input(shape=[len(expanded_training_columns)]),

    keras.layers.Dense(18, activation='sigmoid'),
    keras.layers.Dense(36, activation='sigmoid'),
    keras.layers.Dense(18, activation='sigmoid'),
    keras.layers.Dense( 9, activation='sigmoid'),

    keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history = model.fit(X_train_scaled, y_train, epochs=50, batch_size=32, verbose=0)
```

```
2025-07-28 01:53:57.966623: E external/local_xla/xla/stream_executor/cuda/cuda_driver.cc:152] failed call to cuInit: INTERNAL:
CUDA error: Failed call to cuInit: UNKNOWN ERROR (303)
```

In [8]:
```
csv_test_path  = '/kaggle/input/titanic/test.csv'

test_df = pd.read_csv(csv_test_path)
treated_test_df = treat_data(test_df, training_columns, target_column, False)

X             = treated_test_df[expanded_training_columns]
X_test        = X.values
X_test_scaled = scaler.transform(X_test)

probabilities = model.predict(X_test_scaled)
predictions   = np.where(probabilities > 0.5, 1, 0)
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
```

In [9]:
```
output = pd.DataFrame({'PassengerId': test_df.PassengerId, 'Survived': predictions.ravel()})
output.to_csv('submission.csv', index=False)
print("Your submission was successfully saved!")
```

```
Your submission was successfully saved!
```