🚢 Titanic Survival Prediction Using Deep Learning

📆 Submission Date: 05/02/2025

👤 Student Name: Hamza Boulahia

🏫 University / Institution: ENSET

1️⃣ Introduction

🚢 The Titanic disaster is one of history’s most infamous shipwrecks. This project aims to **predict the survival of Titanic passengers** using **Artificial Neural Networks (ANNs)**.

🔹 Problem Statement

1. Predict whether a passenger **survived (1) or not (0)** based on their characteristics.
2. Use **historical Titanic data** to train and test an ANN model.

🔹 Why Use Deep Learning?

1. Traditional models (Logistic Regression, Decision Trees) perform well, but **ANNs capture complex patterns** in data.
2. We experiment with **different layers and activation functions** to optimize predictions.

2️⃣ Dataset Description

📌 Data Source

The dataset (`titanic.csv`) contains **passenger information**, including:

- `Pclass` (Ticket Class)
- `Sex` (Male/Female)
- `Age` (Passenger's Age)
- `Fare` (Ticket Fare)
- `Embarked` (Port of Embarkation)
- `Survived` (Target Variable: 0 = No, 1 = Yes)

📌 Data Preprocessing

Before training, the data is **cleaned and preprocessed**:

1. **Handling Missing Values**
    - `Age` → Replaced with the **median age**.
    - `Fare` → Replaced with the **median fare**.
    - `Embarked` → Most frequent category (`mode`).
2. **Encoding Categorical Variables**
    - `Sex` → Converted to **0 (Female), 1 (Male)**.
    - `Embarked` → Encoded as **0, 1, 2**.
3. **Feature Scaling**
    - `Age` , `Fare` , and other numeric features were **normalized** for better training.

3️⃣ Model Architecture

We built an **Artificial Neural Network (ANN)** with the following layers:

Layer	Type	Neurons	Activation
1	Input	5	-
2	Hidden	16	ReLU
3	Hidden	8	ReLU
4	Hidden	4	ReLU
5	Output	1	Sigmoid

📌 Model Compilation

- **Loss Function** → Binary Crossentropy (`binary_crossentropy`)
- **Optimizer** → Adam (`adam`)
- **Metrics** → Accuracy (`accuracy`)

📌 Hyperparameters

- **Epochs** → 50
- **Batch Size** → 32
- **Regularization** → Dropout (0.3)
- **Early Stopping** → Stops training if validation loss stops improving.

4️⃣ Training & Performance

The ANN model was trained for **50 epochs** with **training & validation datasets**.

📌 Model Accuracy

After training, the model achieved:

Dataset	Accuracy
Training	**X.XX%**
Validation	**X.XX%**
Test	**X.XX%**

📊 Below is a graph showing accuracy over epochs:

📉 Below is the loss curve:

5️⃣ Evaluation & Results

📌 Model Performance Metrics

Metric	Score
Test Accuracy	**X.XX%**
Precision	**X.XX%**
Recall	**X.XX%**
F1-Score	**X.XX%**

📌 **Interpretation:**

- **Higher accuracy** means the model correctly classifies survival.
- **Precision & Recall** determine if the model makes correct survival predictions.

6️⃣ Model Deployment

📌 Saving & Loading the Model

The trained model was saved using:

```
model.save("titanic_model.h5")
```

To reload and use for predictions:

```
from tensorflow.keras.models import load_model
model = load_model("titanic_model.h5")
```

📌 Making Predictions

To predict survival of **new passengers**:

```
sample_passenger = X_test[:5]  # Select a sample
predictions = model.predict(sample_passenger)
predicted_classes = (predictions > 0.5).astype("int32")
```

8️⃣ Conclusion

✅ We successfully **built & trained an ANN model** to predict Titanic passenger survival.
✅ The model achieved **high accuracy on test data**.
✅ Deep Learning helps in capturing **complex patterns in data**.
✅ Future improvements can make predictions even **more accurate**.