📘 Assignment: Modern ML & Neural Networks

Objective

The objective of this assignment is to:

- Understand **classical machine learning models**
- Implement **core algorithms from scratch**
- Compare **manual implementations vs library implementations**
- Analyze **model performance using metrics**
- Practice **clean OOP-based coding** and **experimental reporting**

You will work primarily on a **movie completion prediction dataset**, and partially on a **music genre classification dataset**.

⚠️ Academic Integrity

- Plagiarism is strictly prohibited
- You may use libraries **only where allowed**
- All manual implementations must be your own

📂 Datasets

1. **Movie Completion Dataset**

   https://www.kaggle.com/datasets/cmaigrot/movies-and-series-rating-from-imdb

2. **Music Genre Classification Dataset**

   https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

---

🧩 Part 1: Decision Tree & Random Forest

1.1 Decision Tree using Library

As demonstrated in class, use a **Decision Tree classifier from a standard ML library** to predict whether a user completes a movie.

You must:

- Train a Decision Tree model
- Try **at least 3 different hyperparameter combinations**, such as:
  - `max_depth`
  - `min_samples_split`
  - `criterion` (gini / entropy)
- Evaluate each model using:
  - Accuracy
  - F1-score

1.2 Comparison Table (Mandatory)

Create a **tabular comparison** in your report with the following columns:

Model	max_depth	min_samples_split	criterion	Accuracy	F1 Score

---

⭐ Bonus (Optional): Random Forest from Scratch

- Implement the **Random Forest algorithm manually**
- You **may use Decision Trees from a library**, but:
  - Bootstrapping
  - Feature sampling
  - Aggregation (majority voting)

  must be written by you
- Compare:
  - Decision Tree vs Random Forest
  - Accuracy
  - F1 Score

🧠 Part 2: Logistic Regression

2.1 Logistic Regression from Scratch

- Implement **Logistic Regression manually**
- Use:
  - Gradient Descent
  - Sigmoid activation

- Binary cross-entropy loss

- Follow **OOP principles** (use a class)

Evaluate:

- Accuracy
- F1 Score

  on the movie dataset.

2.2 Logistic Regression using Library

- Train Logistic Regression using a standard ML library
- Evaluate:
  - Accuracy
  - F1 Score
- Compare results with your manual implementation

⭐ Bonus (Theory): Ensemble Methods with Logistic Regression

In the report, **theoretically explain**:

> Which ensemble method (Bagging, Boosting, Stacking) would be more suitable when using Logistic Regression for this movie dataset, and why?

Support your answer using:

- Bias-variance intuition
- Dataset characteristics

🧠 Part 3: Neural Networks

3.1 Neural Network from Scratch (Movie Dataset)

- Implement a **small Neural Network from scratch**
- Example architecture:
  - 3 layers
  - 5 neurons per layer
- Use:
  - Forward propagation
  - Backpropagation

- Gradient descent

- Follow **OOP design** (class-based implementation)

Train on:

- Movie completion dataset

Report:

- Accuracy
- F1 Score

3.2 Neural Network using Library (Music Genre Dataset)

- Use a deep learning library (TensorFlow / PyTorch)
- Train a Neural Network on **music genre classification**
- Evaluate:
  - Accuracy
  - F1 Score

3.3 Hyperparameter Tuning (Mandatory)

- Study basic concepts of **hyperparameter tuning**
- Try **at least 5 different hyperparameter combinations**, such as:
  - Learning rate
  - Number of layers
  - Number of neurons
  - Batch size
  - Activation functions

Report:

- Best model configuration
- Corresponding Accuracy & F1 Score

📊 Visualization Requirements (Mandatory)

Wherever model training is involved, include:

- Training loss vs epochs
- Validation loss vs epochs

Plots must be:

- Clearly labeled
- Properly explained in the report

👨‍💻 Coding Guidelines (Strict)

- Any **manual implementation** must:
  - Use **Python classes**
  - Follow **OOP principles**
- Code should be:
  - Clean
  - Modular
  - Well-commented

📄 Submission Format

You must submit a zip file containing:

1️⃣ Jupyter Notebook (`.ipynb`)

- All code cells must be:
  - Executed
  - Outputs visible
- Proper section headings
- No broken cells

2️⃣ Report (separate PDF)

The report must include:

- Explanation of each model
- Hyperparameter choices
- Tables of results
- Graphs and interpretations
- Your **methodology and thought process**

📝 Evaluation Criteria

Component	Weight
Correct implementation	40%
Experimental comparison	20%
Report clarity & explanation	20%
Code quality & OOP usage	10%
Bonus sections	10%