1. SUPERVISED LEARNING

1.1 Classification

A. Traditional Algorithms

📌 Project 1: Email Spam Classifier

Description:
Students build a binary classifier to detect spam emails using classical ML algorithms (Logistic Regression, SVM, k-NN, Naive Bayes).

Dataset:

- SpamBase Dataset (UCI ML Repo)
  https://archive.ics.uci.edu/ml/datasets/spambase

Expected Output:

- Notebook/script showing preprocessing, training, evaluation
- Comparison of algorithms with accuracy scores
- Confusion matrix + ROC curve
- GitHub repo with README explaining the approach

📌 Project 2: Loan Approval Prediction

Description:
Predict whether a loan should be approved using Decision Trees & Ensemble methods (Random Forest, XGBoost, LightGBM, CatBoost).

Dataset:

- Loan Prediction Dataset (Kaggle — free download)
  https://www.kaggle.com/datasets/ninzaami/loan-predication

Expected Output:

- Implementation of multiple ensemble models
- Feature importance charts
- Final model with >80% accuracy

- GitHub repo with training code + report

B. Deep Learning Approaches

CNN Projects

📌 Project 3: Handwritten Digit Recognition (Plain CNN)

Dataset:

- MNIST (built-in in Keras/PyTorch)

Expected Output:

- Build basic CNN
- Achieve >98% accuracy
- Visualization of activations/filters
- GitHub repo with model + evaluation

📌 Project 4: Image Classifier with ResNet

Dataset:

- CIFAR-10 (Keras/PyTorch built-in)

Expected Output:

- Build or use pre-trained ResNet
- Train classifier for 10 categories
- Output charts showing training progress
- GitHub submission

RNN Projects

📌 Project 5: Text Sentiment Analyzer (LSTM/GRU)

Dataset:

- IMDb movie reviews (Keras built-in)

Expected Output:

- Build LSTM & GRU models
- Compare accuracy/performance
- Submit trained model + text demo script

Transformer Projects

📌 Project 6: Question Answering using BERT

Dataset:

- SQuAD v1.1 (free)
  https://rajpurkar.github.io/SQuAD-explorer/

Expected Output:

- Fine-tune a BERT QA model
- Evaluate on sample questions
- Submit script to run QA from command line

(If GPU is unavailable, limit training to small batches or use DistilBERT.)

📌 Project 7: Text Generation using GPT-2 (small)

Dataset:

- Students choose any free text corpus (news, Wikipedia subset)

Expected Output:

- Fine-tune GPT-2 small model
- Generate paragraphs based on prompts
- GitHub repository with inference script

1.2 Regression

Traditional Algorithms

📌 **Project 8: House Price Predictor**

Dataset:

- Boston Housing Dataset (Scikit-learn built-in)
  or
- California Housing Dataset (also built-in)

Expected Output:

- Linear & Polynomial regression
- Tree-based regressors
- RMSE & MAE comparison
- GitHub README with evaluation

Deep Learning for Regression

📌 **Project 9: Stock Price Prediction with MLP**

Dataset:

- Yahoo Finance free CSV (any stock — students choose)

Expected Output:

- Build regression MLP
- Predict next-day closing price
- Plot predicted vs actual values
- Git repo with preprocessing + model

---

2. UNSUPERVISED LEARNING

2.1 Traditional Algorithms

📌 Project 10: Customer Segmentation using Clustering

Dataset:

- Mall Customers Dataset (Kaggle — free)
  https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial

Expected Output:

- K-Means, Hierarchical, DBSCAN
- Visualize clusters
- Provide business interpretation
- GitHub code + plots

📌 Project 11: Image Compression using PCA

Dataset:

- Use any set of free images (CIFAR-10 or custom downloaded images)

Expected Output:

- Apply PCA to reduce dimensions
- Reconstruct images
- Compare compressed vs original images
- GitHub repo with scripts

2.2 Deep Unsupervised Learning

📌 Project 12: Autoencoder for Noise Removal

Dataset:

- MNIST or Fashion-MNIST

Expected Output:

- Add noise to images
- Train autoencoder to recover clean images
- Submit before/after comparison plots

📌 Project 13: GAN — Generate Handwritten Digits

Dataset:

- MNIST

Expected Output:

- Build simple GAN
- Train until it generates recognizable digits
- Save generated images to GitHub

---

3. SEMI-SUPERVISED LEARNING

📌 Project 14: Pseudo-Labeling on CIFAR-10

Dataset:

- CIFAR-10 (Keras/PyTorch built-in)

Expected Output:

- Train initial small labeled model
- Generate pseudo-labels for unlabeled data
- Retrain improved model
- Submit accuracy improvement report + code

---

4. REINFORCEMENT LEARNING

📌 Project 15: FrozenLake Agent (Q-Learning)

Dataset / Environment:

- OpenAI Gym — FrozenLake-v1 (free)

Expected Output:

- Train tabular Q-learning agent

- Plot reward curve
- Demonstrate solved environment
- Submit Q-table + training code

📌 Project 16: DQN — CartPole Balancing

Dataset / Environment:

- OpenAI Gym — CartPole-v1

Expected Output:

- Build DQN from scratch
- Achieve ≥200 average reward
- Provide training graph
- Git repo with model & saved weights