# Convolutional Neural Network (CNN) for Image Classification

📄 Description:  
This project implements a CNN using TensorFlow and Keras for image classification on the CIFAR-10 dataset. It involves preprocessing, model building, training, and evaluation.  
  
🎯 Objective:  
To classify images into one of 10 categories using a deep convolutional neural network.  
  
🛠️ Technologies:  
Python, TensorFlow, Keras, Matplotlib, NumPy  
  
🧠 Model:  
- CNN with Conv2D, MaxPooling, Flatten, Dense  
- Accuracy: ~82%  
- Optimizer: Adam  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Object Detection with OpenCV and Haar Cascades

📄 Description:  
A computer vision project for detecting objects like faces and eyes using Haar Cascade Classifiers with OpenCV.  
  
🎯 Objective:  
To detect specific objects (e.g., faces) in real-time using a webcam feed.  
  
🛠️ Technologies:  
Python, OpenCV, Haar Cascade XML classifiers  
  
🧠 Model:  
- Haar Feature-based Cascade Classifier  
- Real-time detection using video capture  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Gemini AI Prompt Engineering (LLM interaction)

📄 Description:  
An experimental notebook demonstrating LLM interaction (e.g., Gemini or GPT) with prompt engineering.  
  
🎯 Objective:  
To explore prompt formatting and control output behavior for generative AI.  
  
🛠️ Technologies:  
Python, Google Generative AI APIs, prompt design techniques  
  
🧠 Model:  
- Large Language Model (e.g., Gemini/GPT)  
- Output customization via prompt manipulation  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# NumPy Tutorial – Scientific Computing in Python

📄 Description:  
A full walkthrough of NumPy, including arrays, broadcasting, indexing, and math operations.  
  
🎯 Objective:  
To master the core concepts of NumPy for numerical computations.  
  
🛠️ Technologies:  
Python, NumPy, Matplotlib  
  
🧠 Concepts:  
- Array manipulation, slicing, broadcasting  
- Mathematical functions and plotting  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Recurrent Neural Network (RNN) for Sequential Prediction

📄 Description:  
An RNN implementation for sequence modeling tasks like time-series or language prediction.  
  
🎯 Objective:  
To predict sequences using RNN layers and understand temporal data modeling.  
  
🛠️ Technologies:  
Python, TensorFlow, Keras, NumPy  
  
🧠 Model:  
- RNN/LSTM layers  
- Use case: next character prediction  
- Metrics: Accuracy, Loss  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Customer Classification using Machine Learning

📄 Description:  
This project performs classification of customers based on features using supervised ML models.  
  
🎯 Objective:  
To categorize customers into different groups (e.g., likely to buy or not).  
  
🛠️ Technologies:  
Python, Pandas, Scikit-learn, Matplotlib  
  
🧠 Model:  
- Logistic Regression, SVM, Decision Tree  
- Accuracy: Up to 92%  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Text Classification using Machine Learning and NLP

📄 Description:  
A machine learning-based pipeline to classify text into categories.  
  
🎯 Objective:  
To build a model that can classify raw text using NLP preprocessing and vectorization.  
  
🛠️ Technologies:  
Python, NLTK, Scikit-learn, Pandas  
  
🧠 Model:  
- Naive Bayes, Logistic Regression, SVM  
- Vectorization: TF-IDF  
- Accuracy: ~85–92%  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# OpenCV – Image Processing Fundamentals

📄 Description:  
Hands-on project showcasing OpenCV techniques for image transformation and edge detection.  
  
🎯 Objective:  
To apply basic OpenCV methods for image manipulation and filtering.  
  
🛠️ Technologies:  
Python, OpenCV, NumPy  
  
🧠 Operations:  
- Canny Edge Detection, Thresholding, Morphological transforms  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Fashion MNIST Image Classification

📄 Description:  
Classify fashion items using a feedforward neural network.  
  
🎯 Objective:  
To implement a neural network for classifying grayscale images into fashion categories.  
  
🛠️ Technologies:  
Python, TensorFlow, Keras, Matplotlib  
  
🧠 Model:  
- Sequential NN with Dense layers  
- Accuracy: ~88–90%  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]

# Stock Price Prediction using LSTM

📄 Description:  
Uses LSTM neural networks to predict stock prices based on historical data.  
  
🎯 Objective:  
To build a deep learning time series model using LSTM.  
  
🛠️ Technologies:  
Python, TensorFlow, Pandas, Matplotlib  
  
🧠 Model:  
- LSTM for time series prediction  
- MSE, loss curves, trend visualization  
  
🔗 GitHub: [Your GitHub Link Here]  
🔗 Colab: [Your Colab Link Here]