1. Data pre-processing operations such as outliers and/or inconsistent data value management.

    a. Load the Titanic dataset from a CSV file or another data source into a Pandas DataFrame.
    b. Identify and handle missing values in the dataset
    c. Create new features if necessary, e.g., 'FamilySize' by combining 'SibSp' and 'Parch.'
    d. Encode categorical variables like 'Sex' and 'Embarked' as numerical values using one-hot encoding or label encoding.
    e. Check for and remove duplicate rows if they exist in the dataset.
    f. Detect outliers (Z-Score, DBSCAN) in numerical columns such as 'Fare' or 'Age' Also analyze the outliers generated by different methods.

2. Implement Feed forward neural network with three hidden layers for classification on CIFAR-10 dataset.
    a. Write code to load the CIFAR-10 dataset and display a few sample images.
    b. Display the dimensions of the images and the number of classes in the dataset.
    c. Implement data preprocessing steps, including normalizing pixel values and one-hot encoding the labels
    d. Implement the architecture of a Feedforward Neural Network with three hidden layers
    e. Compile, Train and evaluate the model performance using metrics such as accuracy, loss, by applying different optimizers (SGD, Adam) and loss functions.

3. Implement a feed forward neural network with three hidden layers for the CIFAR-10 dataset. Record the accuracy and loss during training.
    a. Repeat the training process with Xavier initialization for weight initialization. Compare the convergence speed and accuracy of the network with the baseline results.
    b. Analyze the impact of Xavier initialization on the network's performance.
    c. Repeat the training process with Kaiming initialization for weight initialization. Compare the convergence speed and accuracy of the network with the baseline results.
    d. Analyze the impact of Kaiming initialization on the network's performance.

4. Implement a feed forward neural network with three hidden layers for the CIFAR-10 dataset. Record the accuracy and loss during training.
    a. Implement dropout regulariza-tion by applying dropout to the hidden layers of the network.
    b. Train the network with dropout regularization and compare its performance with the baseline results.
    c. Analyze the impact of dropout on the network's performance in terms of accuracy and overfit-ting

5. Implement a feed forward neural network with three hidden layers for the CIFAR-10 dataset. Record the accuracy and loss during training.
    a. Implement L1 or L2 regularization techniques by adding a regularization term to the loss function during training.
    b. Train the network with regularization and compare its performance with the baseline results.
    c. Analyze the impact of regularization on the network's performance in terms of accuracy and prevention of overfitting

6. Implement CNN for digit classification using MNIST dataset

    a. Write code to load the MNIST dataset and display a few sample images. Discuss the dimensions of the images and the number of classes in the dataset.
    b. Provide code for preprocessing the MNIST dataset before feeding it into the CNN. Include normalization and reshaping of the input images
    c. Implement the architecture of a Convolutional Neural Network (CNN) for digit classification and train it
    d. Implement code to evaluate the trained CNN on a test set. Calculate and display relevant evaluation metrics such as accuracy.

7. Implement a simple RNN for review classification using IMDB dataset
    a. Load the IMDB dataset using the Keras library.
    b. Preprocess the data by limiting the number of words (e.g., max_features=10,000) and setting a maximum sequence length for reviews (e.g., max_len=500).
    c. Include a SimpleRNN layer with an appropriate number of units.
    d. Add a Dense layer with a single neuron and a sigmoid activation function for binary classification.
    e. Train the RNN model on the training dataset for a reasonable number of epochs (e.g., 5 to 10).
    f. Evaluate the trained model on the testing dataset and report the test accuracy.

8. Implement a LSTM for review classification using IMDB dataset
    a. Load the IMDB dataset with only the top 10,000 most frequent words
    b. Pad or truncate the review sequences to a fixed length of 500 words
    c. Create lists to store model performance results
    d. Compile and train the model
    e. Evaluate the model on the test data
    f. Visualize the performance change
    g. Print the test loss and accuracy for different models.

9. Implement a GRU for review classification using IMDB dataset
    a. Load the IMDB dataset with only the top 10,000 most frequent words
    b. Pad or truncate the review sequences to a fixed length of 500 words
    c. Create lists to store model performance results
    d. Compile and train the model
    e. Evaluate the model on the test data
    f. Visualize the performance change
    g. Print the test loss and accuracy for different models.

10. Implement time series forecasting prediction for NIFTY-50 dataset.
    a. Load appropriate dataset
    b. Scale the dataset and convert into time series using TimeseriesGenerator
    c. Implement SimpeRNN with activation function 'tanh', loss function 'MSE', optimizer 'Adam'
    d. Fit the model and evaluate the model on the test data
    e. Print the prediction at any time

11. Implement time series forecasting prediction for Google_Stock_Price dataset.
    a. Load appropriate dataset
    b. Scale the dataset and convert into time series using TimeseriesGenerator
    c. Implement SimpeRNN with activation function 'tanh', loss function 'MSE', optimizer 'Adam'
    d. Fit the model and evaluate the model on the test data
    e. Print the prediction at any time

12. Implement time series forecasting prediction for Sunspots dataset.
    a. Load appropriate dataset
    b. Scale the dataset and convert into time series using TimeseriesGenerator
    c. Implement SimpeRNN with activation function 'tanh', loss function 'MSE', optimizer 'Adam'
    d. Fit the model and evaluate the model on the test data
    e. Print the prediction at any time