Assignment Questions on Phase 1 Traning

1. Numpy and Pandas (Medium)
   - Load the Iris flower dataset using pandas and perform the following operations:
     - Get descriptive statistics of each feature.
     - Visualize the distribution of each feature using histograms and boxplots.
     - Calculate the correlation matrix and identify highly correlated features.
   - Dataset: https://archive.ics.uci.edu/dataset/53/iris
2. Data Preprocessing (Medium)
   - Load the MNIST handwritten digit dataset and perform the following pre-processing steps:
     - Normalize the pixel values of the images.
     - Apply one-hot encoding to the target labels.
     - Split the data into training, validation, and test sets.
   - Dataset: https://github.com/iamavieira/handwritten-digits-mnist
3. Classification (Hard)
   - Build a logistic regression model to classify handwritten digits from the MNIST dataset.
     - Evaluate the model performance using accuracy, precision, recall, and F1 score.
     - Fine-tune the model hyperparameters using grid search CV to improve performance.
     - Visualize the decision boundary of the model.
4. Clustering (Medium)
   - Apply K-means clustering to group customers based on their purchase history and demographic information.
     - Determine the optimal number of clusters using the elbow method.
     - Analyze the characteristics of each cluster to identify customer segments.
     - Visualize the clusters using scatter plots and dimensionality reduction techniques.
   - Dataset: https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales
5. Regression (Hard)

- Build a linear regression model to predict house prices based on features like location, size, and number of bedrooms.
    - Evaluate the model performance using mean squared error (MSE) and R-squared.
    - Perform feature selection to remove irrelevant or redundant features.
    - Interpret the coefficients of the model to understand the importance of each feature.
- Dataset: https://www.kaggle.com/datasets/vikrishnan/boston-house-prices

6. Image Feature Extraction (Hard)

- Extract features from the CIFAR-10 image dataset using the appropriate method
    - Train a simple classifier on the extracted features for image classification.
- Dataset: https://www.vision.caltech.edu/datasets/

7. Text Feature Extraction (Medium)

- Preprocess and extract features from a collection of text reviews.
    - Remove stop words and punctuation.
    - Apply stemming or lemmatization to normalize words.
    - Use TF-IDF or word embedding techniques to represent the text as numerical vectors.
    - Train a sentiment analysis model to classify reviews as positive, negative, or neutral.
- Dataset: https://www.kaggle.com/datasets/marklvl/sentiment-labelled-sentences-data-set

8. CNN for Image Classification (Hard)

- Build a convolutional neural network to classify images of cats and dogs.
    - Design the CNN architecture using convolutional layers, pooling layers, and fully connected layers.
    - Train the model on the labeled dataset and monitor its performance on the validation set.
    - Visualize the filters learned by the convolutional layers to understand how they detect features.
- Dataset: https://www.tensorflow.org/datasets/catalog/oxford_iiit_pet

9. CNN for Text Classification (Hard)

- Build a convolutional neural network for text classification using a pre-trained word embedding model.
    - Use word embeddings to represent words as numerical vectors.

    - Apply convolutional filters to capture local patterns in the text sequences.
    - Train the model on a text classification dataset like sentiment analysis or spam filtering.
- Dataset: https://nlp.stanford.edu/sentiment/

10. Combining Techniques (Hard)

- Combine multiple machine learning techniques to solve a complex problem.
    - Extract features using image and text processing techniques.
    - Train separate classification models for different aspects of the problem.
    - Combine the predictions from the individual models using ensemble methods.
- Dataset: https://paperswithcode.com/task/multimodal-sentiment-analysis