Machine Learning Lab cycle

Question 1: Data Preprocessing (Medium)

a. Load the MNIST handwritten digit dataset and perform the following pre-processing steps:
b. Normalize the pixel values of the images.
c. Apply one-hot encoding to the target labels.
d. Split the data into training, validation, and test sets.
e. Dataset: https://github.com/iamavieira/handwritten-digits-mnist

Question 2: Classification (Hard)

a. Build a logistic regression model to classify handwritten digits from the MNIST dataset.
b. Evaluate the model performance using accuracy, precision, recall, and F1 score.
c. Fine-tune the model hyperparameters using grid search CV to improve performance.
d. Visualize the decision boundary of the model.

Question 3: Clustering (Medium)

a. Apply K-means clustering to group customers based on their purchase history and demographic information.
b. Determine the optimal number of clusters using the elbow method.
c. Analyze the characteristics of each cluster to identify customer segments.
d. Visualize the clusters using scatter plots and dimensionality reduction techniques.
e. Dataset: https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales

Question 4: Feature Selection:

The Iris dataset consists of 150 samples with 4 features each: sepal length, sepal width, petal length, and petal width. Each sample belongs to one of three classes: setosa, versicolor, or virginica.

a. Load the Iris dataset and split it into features (X) and target labels (y).
b. Perform exploratory data analysis (EDA) to gain insights into the dataset.
c. Implement feature selection techniques:
    a. Univariate Feature Selection
    b. Feature Importance using Random Forest
    c. Recursive Feature Elimination (RFE) using Support Vector Machine (SVM)
d. Evaluate the performance of the selected features using a classification model (e.g., SVM or Logistic Regression).
e. Compare the model performance before and after feature selection.

Question 5: Association Rule Mining: The objective of this lab exercise is to understand and implement association rule mining techniques to uncover patterns and associations within a given dataset of transactional data.

B. Dataset:

    a. Take any dataset containing transactions from a grocery store. Each transaction consists of a list of items purchased by a customer. Utilizing association rule mining techniques, analyze the dataset to uncover patterns and associations between items purchased together.
    b. Load the provided dataset containing transactional data from a grocery store.
    c. Examine the structure of the dataset to understand the transactional format.

C. Generating Itemsets:
    a. Implement a function to generate individual itemsets from the dataset.
    b. Calculate the support for each itemset, which represents the frequency of occurrence of each item.

D. Identifying Frequent Itemsets:
    a. Implement the Apriori algorithm to identify frequent itemsets within the dataset.
    b. Determine the minimum support threshold for identifying frequent itemsets.

E. Deriving Association Rules:
    a. Utilize the frequent itemsets obtained from the Apriori algorithm to derive association rules.
    b. Calculate the confidence for each association rule, representing the likelihood of one item being purchased given the purchase of another item.
    c. Set a minimum confidence threshold for selecting meaningful association rules.

F. Evaluation of Association Rules:
    a. Evaluate the generated association rules using appropriate metrics such as support, confidence, and lift.
    b. Identify high-confidence rules with significant lift values, indicating strong associations between items.

Question 6: Collaborative Filtering and Recommender Systems: The objective of this lab exercise is to understand and implement collaborative filtering techniques for building recommender systems.

A. Understanding Collaborative Filtering:
    a. Provide an overview of collaborative filtering and its importance in building recommender systems.
    b. Explain the concepts of user-item interactions, user-based collaborative filtering, and item-based collaborative filtering.

B. Dataset Exploration:
    a. Select a suitable dataset for building a recommender system (e.g., movie ratings, book reviews, product ratings).
    b. Load the dataset and explore its structure and attributes.

C. User-Based Collaborative Filtering:
    a. Implement a user-based collaborative filtering algorithm to recommend items to users based on similarities between users.
    b. Discuss different similarity metrics such as cosine similarity, Pearson correlation, and Euclidean distance.
    c. Evaluate the performance of the user-based collaborative filtering approach using appropriate evaluation metrics (e.g., precision, recall, F1-score).

D. Item-Based Collaborative Filtering:
    a. Implement an item-based collaborative filtering algorithm to recommend items to users based on similarities between items.

   b. Discuss the advantages and disadvantages of item-based collaborative filtering compared to user-based collaborative filtering.
   c. Evaluate the performance of the item-based collaborative filtering approach using similar evaluation metrics as in the user-based approach.

E. Hybrid Approaches:
   a. Discuss hybrid recommender systems that combine collaborative filtering with other techniques such as content-based filtering or matrix factorization.
   b. Implement a simple hybrid recommender system by combining user-based and item-based collaborative filtering approaches.
   c. Evaluate the performance of the hybrid recommender system and compare it with the individual approaches.

F. Evaluation and Interpretation:
   a. Analyze the results of the collaborative filtering and hybrid approaches.
   b. Interpret the recommended items and their relevance to users.
   c. Discuss potential improvements and future directions for enhancing the recommender system's performance.

Question 7: Solving Maze Problem using OpenAI Gym Library: The objective of this lab exercise is to understand and implement a reinforcement learning solution to solve a maze problem using the OpenAI Gym library.

A. Introduction to OpenAI Gym:
   a. Provide an overview of the OpenAI Gym library and its functionalities for reinforcement learning tasks.
   b. Explain the concept of environments, agents, actions, and observations in the context of Gym.

B. Setting up the Maze Environment:
   a. Define a custom maze environment using Gym that represents a maze with walls, a start point, and a goal point.
   b. Implement functions to initialize the environment, reset it to the starting state, and render the maze for visualization.

C. Defining Actions and Observations:
   a. Define the possible actions that an agent can take in the maze environment (e.g., move up, down, left, right).
   b. Determine the observations available to the agent at each state (e.g., current position, proximity to walls or goal).

D. Implementing Q-Learning Algorithm:
   a. Implement the Q-learning algorithm to train an agent to navigate through the maze environment.
   b. Define the Q-table to store Q-values for state-action pairs.
   c. Implement the exploration-exploitation trade-off strategy (e.g., epsilon-greedy) to balance exploration and exploitation during training.

E. Training the Agent:
   a. Train the agent using the Q-learning algorithm to learn an optimal policy for navigating the maze.
   b. Monitor the agent's learning progress by tracking rewards obtained during training episodes.
   c. Visualize the learned policy and the agent's trajectory through the maze.

F. Evaluation and Testing:
    a. Evaluate the trained agent's performance by running episodes in the maze environment and measuring its success rate in reaching the goal.
    b. Analyze the agent's behavior and performance under different conditions (e.g., varying maze sizes, obstacle configurations).

G. Extensions and Improvements:
    a. Experiment with different hyperparameters (e.g., learning rate, discount factor) and observe their impact on the agent's learning and performance.
    b. Explore advanced reinforcement learning techniques (e.g., deep Q-learning) for solving more complex maze environments.

Question 8: Parallel Image Processing with OpenMP: The objective of this lab exercise is to understand and implement parallel image processing operations using OpenMP, focusing on loading an image and performing matrix operations on it.

A. Introduction to OpenMP:
    a. Provide an overview of OpenMP and its usage for parallel programming
    b. Explain the concept of parallelism, threads, and parallel regions in OpenMP.

B. Loading Image:
    a. Write a function to load an image from a file into a matrix representation.
    b. Choose a common image format (e.g., JPEG, PNG) and use appropriate libraries (e.g., OpenCV) for image loading.
    c. Display the loaded image for visualization.

C. Image Processing Operations:
    a. Implement the following image processing operations as matrix operations:
        i. Image Blurring: Apply a blur filter to the image using a convolution matrix/kernel.
        ii. Image Sharpening: Apply a sharpening filter to the image using a convolution matrix/kernel.
        iii. Image Edge Detection: Apply an edge detection filter (e.g., Sobel operator) to detect edges in the image.
        iv. Parallelize each of the matrix operations using OpenMP directives (e.g., #pragma omp parallel for) to exploit parallelism.

D. Performance Analysis:
    a. Measure the execution time of each image processing operation with and without parallelization.
    b. Compare the performance improvement achieved by parallelizing the operations using OpenMP.
    c. Analyze the speedup and efficiency achieved by parallelization.

E. Visualization and Output:
    a. Display the processed images after each image processing operation for visualization.
    b. Save the processed images to files for further analysis and comparison.

Question 9: Perform feature extraction on iris dataset using LDA, PCA, TSNE and SVD to reduce it to,

Case 1: 2 features

Case 2: 3 features and apply cross validation in each case and report your inference.

Question 10: Write a C/C++ program to create a thread to print first n natural numbers.

Question 11: Write a program in C/C++ to create a random integer array of size n and evaluate the performance in terms of execution time.

A. Sequential program with functions
   i. Find the sum of elements in an array
   ii. Search a key element in an array
B. A thread based program to partition the array and perform computation in each thread.
   i. Find the sum of elements in an array
   ii. Search a key element in an array

Question 12: Implement K-means,K-medoid and Fuzzy C-means

Question 13: From sklearn, choose all classifiers and compute evaluation metrics of the iris dataset. After computation,choose the best 4 classifiers which gave the best performance. On these do ensemble methods such as bagging,boosting and stacking.