Machine Learning Lab

Day-1

Study Different types of Data structure using python programming language.

Experiment 1: List

Create a Python list containing the names of five countries. Perform the following operations:

a. Add two more countries to the list.

b. Print the third country in the list.

c. Sort the list in alphabetical order.

d. Check if a country "India" is present in the list.

Create a tuple with five elements, each representing the price of a product. Perform the following operations:

a. Find the maximum and minimum price from the tuple.

b. Calculate the total cost of all products.

c. Convert the tuple to a list and add a new product with its price.

d. Calculate the average price of the products.

Experiment 2: Dictionaries

Create a dictionary that represents the population of five cities. Perform the following operations:

a. Add two more cities and their populations to the dictionary.

b. Find the city with the highest population.

c. Check if a city "London" is present in the dictionary.

d. Remove a city and its population from the dictionary.

Create a dictionary that maps the names of students to their respective marks in an exam. Perform the following operations:

a. Add a new student and their marks to the dictionary.

b. Calculate the average marks of all students.

c. Find the student with the highest marks.

d. Sort the dictionary by student names in alphabetical order.

Experiment 3: Stacks and Queues

Implement a stack using a Python list to perform the following operations:

a. Push five elements onto the stack.

b. Pop two elements from the stack.

c. Check if the stack is empty.

Implement a queue using a Python list to perform the following operations:

a. Enqueue five elements into the queue.

b. Dequeue three elements from the queue.

c. Check if the queue is empty.

Experiment 4: Linked Lists

Implement a singly linked list in Python to perform the following operations:

a. Insert five elements at the beginning of the list.

b. Delete an element from the list.

c. Search for a specific element in the list.

d. Print the elements of the list in reverse order.

Experiment 5: Trees

Implement a binary search tree in Python to perform the following operations:

a. Insert five elements into the tree.

b. Search for a specific element in the tree.

c. Delete an element from the tree.

d. Print the tree using in-order traversal.

Find the height of the binary search tree.

Day-2

Study Different Python Libraries:

1. pandas Library:

Load a dataset (Iris dataset: https://www.kaggle.com/datasets/uciml/iris)using pandas and display the first few rows to understand its structure.

Calculate basic statistics (mean, median, standard deviation, etc.) for a numerical column in the dataset.

Perform data filtering to extract rows based on specific conditions (e.g., SepalLengthCm>5.0).

2. Matplotlib Library:

Create a line plot to visualize the trend of a numerical variable over time.

Generate a histogram to understand the distribution of a numerical variable in the dataset.

Create a bar chart to compare the performance of different categories.

Plot a scatter plot to explore the relationship between two numerical variables.

Customize your plots with labels, titles, colors, and styles.

3. Seaborn Library:

Create a box plot to visualize the distribution of a numerical variable across different categories.

Generate a heatmap to explore the correlation between numerical variables.

Customize the appearance of seaborn plots using various parameters.

4. NumPy Library:

Create a NumPy array and perform basic operations like addition, subtraction, and multiplication.

Use NumPy functions to calculate statistical measures like mean, median, and standard deviation.

Reshape and slice NumPy arrays to extract specific data elements.

Perform element-wise operations and broadcasting with NumPy arrays.

Apply mathematical functions (e.g., exponential, logarithm) to NumPy arrays.

5. SciPy Library:

Use SciPy to perform numerical integration for a given mathematical function.

Day-3

Data preprocessing, data exploration, and data preparation

Data Loading:

Load a dataset (Iris dataset: https://www.kaggle.com/datasets/uciml/iris) into your preferred ML environment (Python).
Display the first few rows of the dataset to inspect its structure and content.
Check the dimensions of the dataset (number of rows and columns).
Identify the data types of each column (numeric, categorical, text, etc.).

Data Exploration:

Calculate basic summary statistics for the numeric columns (mean, median, min, max, standard deviation).

Visualize the distribution of numeric features using histograms or box plots.

Explore the frequency distribution of categorical features using bar plots.

Data Preprocessing:

Handle missing values: Identify and handle any missing values in the dataset (e.g., imputation, removal).

Encode categorical variables: Convert categorical variables into numerical form (e.g., one-hot encoding, label encoding).

Feature scaling: Normalize or standardize numeric features to bring them to a similar scale.

Data Preparation for ML:

Split the dataset into training and testing sets (e.g., 80% for training, 20% for testing).

Ensure the data is in the appropriate format for the ML algorithms (e.g., arrays, matrices).

Day-4

Data Collection and Preprocessing: Obtain the relevant dataset for your experiment. Preprocess the data to handle missing values, outliers, and format it in a way that is suitable for the machine learning algorithms. (Preprocessing the dataset is a critical step in any machine learning experiment to ensure that the data is in a suitable format for training the model. In this experiment, we'll preprocess the "Titanic" dataset to handle missing values, encode categorical features, and split the data into training and testing sets.

Day-5

1. **Implement Linear Regression and calculate sum of residual error on the following datasets**
   - **Y=4X +13+s(0,1)**
   - **Y=10sin(X1)+15sin(X2)+s(0,1)**
   - **Boston Housing Rate Dataset**

Where X is random variable with values [0,50] and s(0,1) Gaussian white noise of zero mean and unit variance.

Consider the following instructions

1. **Implement both the analytic matrix formulation and gradient descent on LMS (Least Mean square) loss formulation to compute the weight matrix and compare the results.**
2. **For gradient Descent implement full batch, stochastic and mini batch stochastic formulations and compare the results.**

3. **Consider the different values for learning rate in gradient descent and study their effect on convergence and thus implement decaying learning rate system.**
4. **Plot the data points and the computed regression line for dataset-1and dataset-2.**

The program can be written Python programming language from scratch. No machine learning/data science/statistics package/library should be used.

Day-6

1. **Implement Logistic Regression on Cancer dataset and print the confusion matrix.**
   Dataset: https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset
2. **Apply multivariate regression on Boston Housing Rate Dataset to predict house price from the given fourteen independent variables.**

Day-7

1. **Load the "iris" dataset and perform k-nearest neighbor classification. Plot the accuracy/error w.r.t different k values. Compare the accuracy with the built-in function of k-NN.**

The program can be written Python programming language from scratch. No machine learning/data science/statistics package/library should be used.

Day-8

1. **For a given dataset (e.g. iris data set) D of size N×M with N:Number of samples and M: number of features, design a**

Bayesian classifier/ Support vector machine/ Decision tree to classify the test data. Divide the data set into training or testing data according to random percentage split. (assume the underlying distribution to Gaussian).

Day-9

1. **Implement Principal Component Analysis Algorithm and use it to reduce dimension of iris dataset.**
   Consider the following instructions:
   - **Plot the magnitude of eigen values in sorted order.**
   - **Plot the reconstructed data points along with the class labels using 1 and 2 PCs for reconstruction.**
   - **Classify the dimension reduced dataset using Bayes Classifier.**

Day-10

Write a program to cluster a set of points using K-means. Consider, K=3, clusters. Consider Euclidean distance as the distance measure. Randomly initialize a cluster mean as one of the data points. Iterate for 10 iterations. After iterations are over, print the final cluster means for each of the clusters. Use the ground truth cluster label present in the data set to compute and print the Jacquard distance of the obtained clusters with the ground truth clusters for each of the three clusters.

Day-11

Ensemble Learning:

Write a program to implement the Adaboost algorithm with decision tree as the base classifier. The decision tree implemented as a function. Run Adaboost for 3 rounds. The combined classifier should be tested on test instances and the accuracy of prediction for the test instances should be printed as output. A single program should train the classifier on the training set as well as test it on the test set.

Dataset: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Day-12

Breast cancer is one of the most common cancers among women and men globally. Breast cancer arises when cells in the breast start to develop abnormally. Due to this cancer, there is a huge number of deaths every year. It is the most common type of all cancers and also the main cause of women's death worldwide. So with the help of Machine learning if we can classify the patient having which type of cancer, then it will be easy for doctors to provide timely treatment to patients and improve the chance of survival.

In this Machine learning Lab experiment we are going to analyze and classify Breast Cancer (that the breast cancer belongs to which category), as basically there are two categories of breast cancer that is:

- **Malignant type breast cancer**
- **Benign type breast cancer**

Design a computer aided diagnosis (CAD) model to classify the breast cancer. It involves a 3-step work:

a. Read the dataset

b. Optimal Feature Selection (Apply PCA)

c. Apply different Classifier for classification like KNN (with N=3, 5, 7), Naïve Bayes, Random forest, Decision tree, Support Vector Machine).

d. Reporting metrics (Accuracy, Precision, Recall, F1-score, Matthews correlation coefficient), Draw the ROC curve (True Positive Rate vs. False positive Rate)

e. Compare the classification results.

Dataset: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Day-13

Create a simulated stock trading environment where an agent can buy or sell stocks based on historical price data. Use a reinforcement learning algorithms like Deep Q-Network (DQN) to train the agent to make profitable trading decisions.

…….Best of Luck……