Practical-1

AIM: Study of Machine learning basics

In this lab, we will go through the basics of machine learning. The student needs to make a soft copy note on the following topics:

Topics:

1. **What is Machine learning**

- Machine learning is a field of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computer systems to learn from and make predictions or decisions based on data, without being explicitly programmed. The goal of machine learning is to allow computers to automatically analyze and interpret complex patterns and relationships in large datasets, and use that knowledge to make accurate predictions or take appropriate actions.

2. **Steps in collection of data**

- 1) Identify the purpose

  2) Set goals

  3) Plan an approach and methods

  4) Collect data

  5) Analyse and interpret data

  6) Act on results

3. **Steps in importing the data in python (Through: csv, json, and other data formats)**

- 1) Importing CSV data:
  - Import the 'csv' module: 'import csv'
  - Open the CSV file using 'open()' and create a CSV reader object.
  - Read the data using the reader object and store it in appropriate data structures (e.g., lists, dictionaries).
  - Close the CSV file.

  <u>Example:</u>

```
import csv
with open('data.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        # Process each row of data
        print(row)
```

2) Importing JSON data:

- Import the ‘json’ module: ‘import json’
- Open the JSON file using ‘open()’ and load the data using ‘json.load()’
- Close the JSON file.

Example:

```
import json
with open('data.json', 'r') as file:
    data = json.load(file)
# Process the data
print(data)
```

3) Importing data from other formats:

Excel files (.xlsx): You can use the pandas library, which provides the read_excel() function.

4. Preprocessing

a) Remove Outliers

Outliers are data points that deviate significantly from the majority of the dataset. They can distort analysis results and affect the performance of machine learning models. To remove outliers, you can use various techniques such as:

Z-Score: Calculate the z-score for each data point and remove the ones that fall outside a certain threshold (e.g., z-score greater than 3 or less than -3).

Interquartile Range (IQR): Calculate the IQR for a feature and remove data points that lie below the lower quartile minus 1.5 times the IQR or above the upper quartile plus 1.5 times the IQR.

b) Normalize Datasets, Data encoding

Normalization is the process of scaling numerical data to a common range. It ensures that all features contribute equally during analysis or model training. Common normalization techniques include:

Min-Max Scaling: Scale the data to a specific range, often between 0 and 1, using the formula: (x - min) / (max - min).

Standardization: Transform the data to have a mean of 0 and a standard deviation of 1 using the formula: (x - mean) / standard deviation.

Data encoding is necessary when dealing with categorical or text data that machine learning algorithms cannot directly process. Some common encoding techniques are:

One-Hot Encoding: Create binary columns for each category and represent the presence or absence of a category with 1s and 0s, respectively.

Label Encoding: Assign a unique numerical label to each category. This encoding is suitable for ordinal categorical data where the order matters.

c) Handling Missing Data

Missing data can occur due to various reasons and can cause issues in analysis and modeling. Strategies to handle missing data include:

Removal: Remove rows or columns with missing values. This approach is suitable if the missing values are minimal and won't significantly impact the analysis.

Imputation: Fill in the missing values with estimated or substituted values. Common imputation techniques include mean, median, mode, or using more advanced methods like regression imputation or k-nearest neighbors imputation.

5. Machine Models

a) Types of machine learning models – Supervised learning, Unsupervised learning, reinforcement learning.

1) Supervised Learning:

Supervised learning involves training a model on labeled data, where the input data is paired with corresponding target labels or outputs. The goal is to learn a mapping function that can predict the correct labels for new, unseen data. Supervised learning can be further divided into two subcategories:

- Classification: In classification, the target variable is categorical, and the model's objective is to assign input data points to predefined classes or categories. Example algorithms include logistic regression, support vector machines (SVM), and random forests.

- Regression: In regression, the target variable is continuous or numerical, and the model aims to predict a value or quantity. Example algorithms include linear regression, decision trees, and neural networks.

2) Unsupervised Learning:

Unsupervised learning involves training a model on unlabeled data, where there are no predefined target labels or outputs. The objective is to discover patterns, relationships, or structures in the data. Unsupervised learning can be further divided into two subcategories:

- Clustering: Clustering algorithms group similar data points together based on their intrinsic characteristics or proximity. Examples include k-means clustering, hierarchical clustering, and DBSCAN.
- Dimensionality Reduction: Dimensionality reduction techniques aim to reduce the number of features or variables in the data while preserving its essential information. It helps in visualizing and compressing high-dimensional data. Examples include principal component analysis (PCA)

3) Reinforcement Learning:

Reinforcement learning involves an agent learning to interact with an environment and make decisions to maximize rewards or minimize penalties. The agent learns through trial and error, receiving feedback from the environment in the form of rewards or punishments. Reinforcement learning is commonly used in scenarios where an agent learns to play games, control robotic systems, or optimize complex tasks. Examples include Q-learning, policy gradients

b) Parameters of machine learning model (Learning rate, regularization, etc.)

- Learning Rate:

  The learning rate determines the step size at which a model's parameters are updated during training. It controls the speed at which the model learns from the data. A high learning rate may result in faster convergence but can cause overshooting, while a low learning rate may lead to slow convergence or getting stuck in local optima. It is typically set prior to training and can be adjusted to achieve better performance.

- Regularization:

  Regularization techniques are used to prevent overfitting, which occurs when a model learns too much from the training data and performs poorly on unseen data. Regularization adds a penalty term to the model's loss function to discourage complex or large parameter values. Common regularization techniques include:

  L1 Regularization (Lasso): Adds the absolute value of the parameter coefficients to the loss function.

  L2 Regularization (Ridge): Adds the squared value of the parameter coefficients to the loss function.

  Dropout: Randomly sets a fraction of the model's inputs or neurons to zero during training to reduce dependence on specific features.

6. **Test-train data split: using constant ration, k-fold cross validation**

    - Constant Ratio Test-Train Split:

      In this approach, a fixed ratio or percentage of the data is allocated to the training set, and the remaining portion is allocated to the testing set. The typical ratio is 70-30 or 80-20, but it can vary based on the size of the dataset and the specific problem. The data is randomly shuffled before the split to ensure randomness in the distribution of samples between the training and testing sets. The model is trained on the training set and evaluated on the testing set to assess its performance.

    - K-Fold Cross-Validation:

      K-fold cross-validation is a technique that divides the data into k equally sized folds or subsets. The model is trained and evaluated k times, each time using a different fold as the testing set and the remaining folds as the training set. The performance metrics from each fold are then averaged to obtain an overall performance measure. This technique helps in obtaining a more robust estimate of the model's performance by using multiple train-test splits. Common values for k are 5 or 10, but it can be adjusted based on the dataset size and computational resources.

7. **Output Inference**

    Output inference refers to the process of making predictions or inferences based on the output of a trained model. Once a model has been trained on a dataset, it can be used to generate predictions or infer insights from new, unseen data. The output inference step is typically performed after the model has been trained and validated.

8. **Validation: different metrics – Confusion Matrix, Precision, Recall, F1-score**

    Validation is an essential step in machine learning to assess the performance and effectiveness of a trained model. Various metrics are used to evaluate model performance, including the confusion matrix, precision, recall, and F1-score. Let's understand each of these metrics:

    1) Confusion Matrix

       Confusion Matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values. It presents the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

    2) Precision

       Precision is a metric that indicates the proportion of correctly predicted positive instances out of all instances predicted as positive. It is calculated as TP / (TP + FP).

Precision focuses on the accuracy of positive predictions and is useful when the cost of false positives is high (e.g., in medical diagnosis).

3) Recall

Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. It is calculated as TP / (TP + FN). Recall emphasizes the model's ability to identify all positive instances and is particularly important when the cost of false negatives is high (e.g., in spam email detection).

4) F1-score

The F1-score is a harmonic mean of precision and recall, providing a single metric that balances both metrics. It is calculated as 2 * (Precision * Recall) / (Precision + Recall). The F1-score is useful when there is an imbalance between the number of positive and negative instances in the dataset.