**Machine Learning Lab Manual**

**Title:-Introduction to how we add Pandas and numpy library in python**

**What is NumPy?**

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since, arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, Tensor Flow, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation.

NumPy’s main object is the homogeneous multidimensional array. It is a table with same type elements, i.e., integers or string or characters (homogeneous), usually integers. In NumPy, dimensions are called axes. The number of axes is called the rank.

**Why Use NumPy?**

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

**Why is NumPy Faster than Lists?**

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

This behaviour is called locality of reference in computer science.

This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

**Installation:** users can install NumPy via pip command:

**pip install numpy**

Numpy can be imported into the notebook using

import numpy as np

**What is Pandas?**

Pandas is a robust, popular, **open-source** Python package that is loaded with data science and data analysis methods andfunctions. It also helps in performing machine learning tasks. This open-source library was created by Wes McKinney in 2008 atAQR Capital Management. **Pandas** library provides flexible, expressive, and fast data structures that exploit numerical data andtime series.

**Why use Pandas?**

Pandas allow making judgments based on **statistical theories** by analyzing big data. In data science, relevant data is very crucial. Pandas make the messy data readable, structured, and relevant by cleaning them up. Pandas show **co-relations** between any two rows and columns. It also presents the average value, maximum value, and minimum value.

**Install Pandas with pip:**

**pip install pandas**

**Program 2-**

**Write a python program to NumPy Arithmetic Operations using Numpy library functions**

**NumPy Arithmetic Operations**

Arithmetic operations are possible only if the array has the same structure and dimensions. We carry out the operations following the rules of array manipulation. We have both functions and operators to perform these functions.

**NumPy Add function**

This function is used to add two arrays. If we add arrays having dissimilar shapes we get “Value Error”.

import numpy as np

a = np.array([10,20,100,200,500])

b = np.array([3,4,5,6,7])

np.add( a,b)

**NumPy Subtract function**

We use this function to output the difference of two arrays. If we subtract two arrays having dissimilar shapes we get “Value Error”

np.subtract(a, b)

**NumPy Multiply function**

We use this function to output the multiplication of two arrays. We cannot work with dissimilar arrays

np.multiply(a, b)

**NumPy Divide Function**

We use this function to output the division of two arrays. We cannot divide dissimilar arrays.

np.divide(a,b)

**Program-3**

**Write a python program for pandas library functions**

Pandas deals with the following three data structures −

* Series
* DataFrame
* Panel

These data structures are built on top of Numpy array, which means they are fast.

Dimension & Description

The best way to think of these data structures is that the higher dimensional data structure is a container of its lower dimensional data structure. For example, Data Frame is a container of Series, Panel is a container of DataFrame.

|  |  |  |
| --- | --- | --- |
| **Data Structure** | **Dimensions** | **Description** |
| Series | 1 | 1D labeled homogeneous array, sizeimmutable. |
| Data Frames | 2 | General 2D labeled, size-mutable tabular structure with potentially heterogeneously typed columns. |
| Panel | 3 | General 3D labeled, size-mutable array. |

Building and handling two or more dimensional arrays is a tedious task, burden is placed on the user to consider the orientation of the data set when writing functions. But using Pandas data structures, the mental effort of the user is reduced.

For example, with tabular data (DataFrame) it is more semantically helpful to think of the **index** (the rows) and the **columns** rather than axis 0 and axis 1.

### **Mutability**

All Pandas data structures are value mutable (can be changed) and except Series all are size mutable. Series is size immutable.

**Note** − DataFrame is widely used and one of the most important data structures. Panel is used much less.

Series

Series is a one-dimensional array like structure with homogeneous data. For example, the following series is a collection of integers 10, 23, 56, …

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 23 | 56 | 17 | 52 | 61 | 73 | 90 | 26 | 72 |

DataFrame

DataFrame is a two-dimensional array with heterogeneous data. For example,

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Age** | **Gender** | **Rating** |
| Steve | 32 | Male | 3.45 |
| Lia | 28 | Female | 4.6 |
| Vin | 45 | Male | 3.9 |
| Katie | 38 | Female | 2.78 |

The table represents the data of a sales team of an organization with their overall performance rating. The data is represented in rows and columns. Each column represents an attribute and each row represents a person.

## Panel

Panel is a three-dimensional data structure with heterogeneous data. It is hard to represent the panel in graphical representation. But a panel can be illustrated as a container of DataFrame.

**Program 4-**

**Program to calculate mean, median, mode of given number of dataset**

* **Mean**: We can define the mean as the average value of all numbers. It is also called the arithmetic mean. To find the average of all numbers, the basic approach or the arithmetic approach is to add all the numbers and divide that addition with the quantity of numbers. Let suppose, you have five numbers (2, 4, 3, 7, 9). To find the average of these numbers, you have to simply add them (2+4+3+7+9) and divide the addition with 5 (because it has five numbers).
* **Median**: The median is the middle value in a cluster of numbers or values. In this, the group of values remains sorted in either ascending or descending order. If there is an odd quantity of numbers, the median value will be in the middle having the same amount of numbersbefore and after it. Suppose we have 2, 3, 4, 5, 6, and then 4 is the median value in this number group.
* **Mode**: We can define mode as that particular number, which occurs most often in a cluster of numbers or values. The mode number will appear frequently, and there can be more than one mode or even no mode in a group of numbers. Suppose we have 3, 4, 7, 4, 2, 8, 6, 2. Then, here are two mode numbers, 4 and 2.

**Program 5-**

**IMPLEMENT LINEAR REGRESSION USING PYTHON.**

DESCRIPTION:

**Regression:** Regression analysis is one of the most important fields in statistics and machine learning. There are many regression methods available. Linear regression is one of them

**What Is Regression?**

Regression analysis is one of the most important fields in statistics and machine learning. There are many regression methods available. Linear regression is one of them. Regression searches for relationships among variables. For example, you can observe several employees of some company and try to understand how their salaries depend on the **features**, such as experience, level of education, role, city they work in, and so on. This is a regression problem where data related to each employee represent one **observation**. The presumption is that the experience, education, role, and city are the independent features, while the salary depends onthem.Generally, in regression analysis, you usually consider some phenomenon of interest and there have a number of observations. Each observation has two or more features. Following the assumption that (at least) one of the features depends on the others, you try to establish a relation among them. You need to find a function that maps some features or variables to others sufficiently well. The dependent features are called the **dependent variables**, **outputs**, or **responses**. The independent features are called the **independent variables**, **inputs**, or **predictors**.

**Implementing Linear Regression in Python**

It’s time to start implementing linear regression in Python. Basically, all you should do is apply the proper packages and their functions and classes.

**Python Packages for Linear Regression**

The package **scikit-learn** is a widely used Python library for machine learning, built on top of NumPy and some other packages. If you want to implement linear regression and need the functionality beyond the scope of scikit- learn, you should consider **stats models**. It’s a powerful Python package for the estimation of statistical models, performing tests, and more. It’s open source as well.

**Simple Linear Regression With scikit-learn**

Let’s start with the simplest case, which is simple linear regression. There are five basic steps when you’re implementing linear regression:

1. Import the packages and classes you need.

2. Provide data to work with and eventually do appropriate transformations.

3. Create a regression model and fit it with existing data.

4. Check the results of model fitting to know whether the model is satisfactory.

5. Apply the model for predictions.

**Program 6**

Implement Logistic Regression Model

## Logistic Regression

* Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



**Program 7**

**Program for train and test dataset**

**What is Training Dataset?**

The ***training data is the biggest (in -size) subset of the original dataset, which is used to train or fit the machine learning model***. Firstly, the training data is fed to the ML algorithms, which lets them learn how to make predictions for the given task.

The training data varies depending on whether we are using Supervised Learning or Unsupervised Learning Algorithms.

For **Unsupervised learning**, the training data contains unlabeled data points, i.e., inputs are not tagged with the corresponding outputs. Models are required to find the patterns from the given training datasets in order to make predictions.

On the other hand, for supervised learning, the training data contains labels in order to train the model and make predictions.

The type of training data that we provide to the model is highly responsible for the model's accuracy and prediction ability. It means that the better the quality of the training data, the better will be the performance of the model. Training data is approximately more than or equal to 60% of the total data for an ML project.

## What is Test Dataset?

Once we train the model with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. **The test dataset is another subset of original data, which is independent of the training dataset**. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

At this stage, we can also check and compare the testing accuracy with the training accuracy, which means how accurate our model is with the test dataset against the training dataset. If the accuracy of the model on training data is greater than that on testing data, then the model is said to have over fitting.

The testing data should:

* Represent or part of the original dataset.
* It should be large enough to give meaningful predictions.

Splitting the dataset into train and test sets is one of the important parts of data pre-processing, as by doing so, we can improve the performance of our model and hence give better predictability.



Therefore, if we train and test the model with two different datasets, then it will decrease the performance of the model. Hence it is important to split a dataset into two parts, i.e., train and test set.

**Program 8**

**Write a python program to implement clustering Algorithm**.

**Clustering:-**

Clustering is a set of techniques used to partition data into groups, or clusters. **Clusters** are loosely defined as groups of data objects that are more similar to other objects in their cluster than they are to data objects in other clusters. In practice, clustering helps identify two qualities of data:

1. Meaningfulness
2. Usefulness

**Meaningful** clusters expand domain knowledge. For example, in the medical field, researchers applied clustering to gene expression experiments. The clustering results identified groups of patients who respond differently to medical treatments.

**Useful** clusters, on the other hand, serve as an intermediate step in a [data pipeline](https://en.wikipedia.org/wiki/Pipeline_(computing)). For example, businesses use clustering for customer segmentation. The clustering results segment customers into groups with similar purchase histories, which businesses can then use to create targeted advertising campaigns.

The **k-means clustering** method is an [unsupervised machine learning](https://en.wikipedia.org/wiki/Unsupervised_learning) technique used to identify clusters of data objects in a dataset. There are many different types of clustering methods, but k-means is one of the oldest and most approachable. These traits make implementing k-means clustering in Python reasonably straightforward, even for novice programmers and data scientists.

If you’re interested in learning how and when to implement k-means clustering in Python, then this is the right place. You’ll walk through an end-to-end example of k-means clustering using Python, from preprocessing the data to evaluating results.

### **Understanding the K-Means Algorithm**

Conventional k-means requires only a few steps. The first step is to randomly select k centroids, where k is equal to the number of clusters you choose. **Centroids** are data points representing the center of a cluster.

The main element of the algorithm works by a two-step process called **expectation-maximization**. The **expectation** step assigns each data point to its nearest centroid. Then, the **maximization** step computes the mean of all the points for each cluster and sets the new centroid. Here’s what the conventional version of the k-means algorithm looks like:

#### **Implementation of the K-Means Algorithm**

The implementation and working of the K-Means algorithm are explained in the steps below:

**Step 1:** Select the value of K to decide the number of clusters (n\_clusters) to be formed.

**Step 2:** Select random K points that will act as cluster centroids (cluster\_centers).

**Step 3:** Assign each data point, based on their distance from the randomly selected points (Centroid), to the nearest/closest centroid, which will form the predefined clusters.

**Step 4:** Place a new centroid of each cluster.

**Step 5:** Repeat step no.3, which reassigns each datapoint to the new closest centroid of each cluster.

**Step 6:** If any reassignment occurs, then go to step 4; else, go to step 7.

**Step 7:** Finish