UNIT II – REGRESSION AND PREDICTIVE MODELING

Machine learning platforms and development environments – Python ecosystem for ML (NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch) – Dataset splitting: training, validation, testing – Performance metrics: MSE, RMSE, MAE, R^2 score – Linear Regression – Multiple Linear Regression – Polynomial Regression – Ridge Regression – Lasso Regression – Elastic Net – Logistic Regression for binary classification – Gradient Descent optimization – Hyperparameter tuning.

MACHINE LEARNING PLATFORMS AND DEVELOPMENT ENVIRONMENTS

- The Python ecosystem for machine learning is built on a modular stack where specialized libraries handle data computation, classical modelling, and deep learning.
- This unified stack allows developers to transition smoothly from data ingestion to heavy neural network training within a single pipeline

PYTHON ECOSYSTEM FOR ML

(NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch)

- Python libraries are collection of codes and functions that can be used in a program for a specific task.
- They are generally used to ease the process of programming when the tasks are repetitive and complex.
- Machine Learning is an interdisciplinary field where each algorithm is developed on combining programming and mathematics.
- Instead of manually coding the complete algorithm with mathematical and statistical formulas, using libraries would make the task easy.
- Python is the most popular programming language specially to implement machine learning because of its simplicity, vast collection of libraries and easiness.
- Some popular Python machine learning libraries are as follows −
  - NumPy
  - Pandas
  - SciPy
  - Scikit-learn
  - PyTorch
  - TensorFlow
  - Keras
  - Matplotlib
  - Seaborn
  - OpenCV
  - NLTK
  - SpaCy

NumPy

- NumPy is a general purpose array and matrix processing package used for scientific computing and to perform a variety of mathematical operations like linear algebra, Fourier transform and others.
- It provides a high performance multi-dimensional array object and tools , to manipulate the matrices for the improvement of machine learning algorithms.
- It is a critical component of the Python machine learning ecosystem, as it provides the underlying data structure and numerical operations required for many machine learning algorithms
- By using NumPy, we can perform the following important operations −
  - Mathematical and logical operations on arrays.
  - Fourier transformation
  - Operations associated with linear algebra.
- We can also see NumPy as the replacement of MATLAB because NumPy is mostly used along with Scipy (Scientific Python) and Mat-plotlib (plotting library).

Installation and Execution

- If you are using Anaconda distribution, then no need to install NumPy separately as it is already installed with it. You just need to import the package into your Python script with the help of following −
  - import numpy as np
- On the other hand, if you are using standard Python distribution then NumPy can be installed using popular python package installer, pip.
  - pip3 install numpy

Example - Using Array

```
import numpy as np
data = np.array([1,2,3,4,5])
print(data)
print(len(data))
print(type(data))
print(data.shape)
```

Output

The above Python example code will produce the following result −

```
[1 2 3 4 5]
5
<class 'numpy.ndarray'>
(5,)
```

Pandas

- Pandas is a powerful library for data manipulation and analysis.
- This library is not exactly used in machine learning algorithms but is used in the prior step i.e., for data preparation.

- It functions based on two data structures:
  - Series(one-dimensional)
  - Data frames(two-dimensional).
- This allows it to handle vast typical use cases in various sectors like Finance, Business, and Health
- With the help of Pandas, in data processing, we can accomplish the following five steps −
  - Load
  - Prepare
  - Manipulate
  - Model
  - Analyze

Data Representation in Pandas

Series − It is a one-dimensional ndarray with an axis label, which means it is like a simple array with homogeneous data.

For example, the following series is a collection of integers 1,5,10,15,24,25...

1 5 10 15 24 25 28 36 40 89

Data frame − It is the most useful data structure and is used for almost all kinds of data representation and manipulation in pandas.

- It is a two-dimensional data structure that can contain heterogeneous data.
- Generally, tabular data is represented by using data frames.

For example, the following table shows the data of students having their names and roll numbers, age and gender −

Name	Roll number	Age	Gender
Aarav	1	15	Male
Harshit	2	14	Male
Kanika	3	16	Female
Mayank	4	15	Male

Panel − It is a 3-dimensional data structure containing heterogeneous data.

- It is very difficult to represent the panel in graphical representation, but it can be illustrated as a container of DataFrame.
- The following table gives us the dimension and description about the above-mentioned data structures used in Pandas −

We can understand these data structures as the higher dimensional data structure is the container of lower dimensional data structure.

Data Structure	Dimension	Description
Series	1-D	Size immutable, 1-D homogeneous data
DataFrames	2-D	Size Mutable, Heterogeneous data in tabular form
Panel	3-D	Size-mutable array, container of DataFrame.

Installation and Execution

- If you are using Anaconda distribution, then no need to install Pandas separately as it is already installed with it. You just need to import the package into your Python script with the help of following −
  - import pandas as pd
- On the other hand, if you are using standard Python distribution then Pandas can be installed using popular python package installer, pip.
  - pip3 install pandas
- After installing Pandas, you can import it into your Python script

Example - Series of Array

The following is an example of creating a series from ndarray by using Pandas −

```
import pandas as pd
import numpy as np
data = np.array(['g','a','u','r','a','v'])
s = pd.Series(data)
print (s)
```

Output
The above example code will produce the following result −

- 0 g
- 1 a
- 2 u
- 3 r
- 4 a
- 5 v
- dtype: object

SciPy

- SciPy is an open-source library that performs scientific computing on large datasets.
- It is easy to use and fast to execute data visualization and manipulation tasks.
- It consists of modules used for the optimization of algorithms and to perform operations like integration, linear algebra, or signal processing.
- SciPy is built on NumPy but extends its functionality by performing complex tasks like numerical algorithms and algebraic functions.

Installation and Execution

- If you are using Anaconda distribution, then no need to install SciPy separately as it is already installed with it. You just need to use the package into your Python script.
- For example, with the following line of script we are importing linalg submodule from scipy −
  - from scipy import linalg

- On the other hand, if you are using standard Python distribution and having NumPy, then SciPy can be installed using a popular python package installer, pip.
  - pip3 install scipy

Example - Matrix Operations

Following is an example of creating a two-dimensional array (matrix) and finding the inverse of the matrix.

```
import numpy as np
import scipy
from scipy import linalg
A= np.array([[1,2],[3,4]])
print(linalg.inv(A))
```

Output

The above Python example code will produce the following result −

```
[[-2.   1. ]
[ 1.5 -0.5]]
```

Scikit-learn

- Scikit-learn, a popular open-source library built on NumPy and SciPy, is used to implement machine learning models and statistical modeling.
- It supports supervised and unsupervised learning. It provides various tools for implementing data pre-processing, feature selection, model selection, model evaluation, and many other tasks.

Features

- It is built on NumPy, SciPy, and Matplotlib.
- It is an open source and can be reused under BSD license.
- It is accessible to everybody and can be reused in various contexts.
- Wide range of machine learning algorithms covering major areas of ML like classification, clustering, regression, dimensionality reduction, model selection etc. can be implemented with the help of it.

Installation and Execution

- If you are using Anaconda distribution, then there is no need to install Scikit-learn separately as it is already installed with it. You just need to use the package into your Python script.
- For example, with the following line of the script, we are importing a dataset of breast cancer patients from Scikit-learn −
  - from sklearn.datasets import load_breast_cancer
- On the other hand, if you are using standard Python distribution and having NumPy and SciPy, then Scikit-learn can be installed using the popular python package installer, pip.
  - pip3 install scikit-learn
- After installing Scikit-learn, you can use it in your Python script as you have done above.

Example - Using Breast Cancer DataSet

```
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
print(data.target[[10, 50, 85]])
print(list(data.target_names))
```

Output
The above python example code will produce the following result −

```
[0 1 0]
['malignant', 'benign']
```

PyTorch

- PyTorch is an open-source Python library based on Torch library, generally used for developing deep neural networks.
- It is based on intuitive Python and can dynamically define computational graphs.
- PyTorch is particularly useful for researchers and developers who need a flexible and powerful deep learning framework.

Installation and Execution

- For Python 3.8 or later and CPU plateform on Windows operating system, you can use the following command to install PyTorch (torch, torchvision and torchaudio)
  - pip3 install torch torchvision torchaudio
- To import PyTorch use the following −
  - import torch
- After installing PyTorch, you can import it into your Python script

Example - Using PyTorch Tensor

Following is an example of creating a NumPy array and converting it to a PyTorch tensor −

```
import numpy as np
import torch
x = np.ones([3,4])
y = torch.from_numpy(x)
print(y)
```

Output
The above example code will produce the following result −

```
tensor([[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]], dtype=torch.float64)
```

TensorFlow

- TensorFlow is one of the most known software libraries developed by Google to implement machine learning and deep learning tasks.
- The creation of computational graphs and efficient execution on various hardware platforms is made easier with this.
- It is widely used for the development of tasks like natural language processing, image recognition and handwriting recognition.

Installation and Execution

- For CPU platform on Windows operating system, you can use the following command to install TensorFlow using pip −
  - pip3 install tensorflow
- To import TensorFlow use the following −
  - import tensorflow as tf
- After installing TensorFlow, you can import it into your Python script

Example - Using TensorFlow

```
import tensorflow as tf
data = tf.constant([[2,1],[4,6]])
print(data)
```

Output

- The above example code will produce the following result
- tf.Tensor(
- [[2 1]
- [4 6]], shape=(2, 2), dtype=int32)

Matplotlib

- Matplotlib is a popular plotting library usually used for data visualization, to create graphs, plots, histograms and bar charts.
- It provides tools and functions for data analysis, exploration and presentation tasks.

Installation and Execution

- We can use the following line of script to install Matplotlib using pip −
  - pip3 install matplotlib
- Most of the matplotlib utilities lies under the pyplot submodule. We can import pyplot from Matplot using the following lines of script −
  - import matplotlib.pyplot as plt
- After installing Matplotlib, you can import it into your Python script

Example - Plotting a Straight Line

```
import matplotlib.pyplot as plt
plt.plot([1,2,3],[1,2,3])
```

```
plt.show()
```
Slide 33

Seaborn

- Seaborn is an open-source Python library built based on Matplotlib and integrates with Pandas.
- It is used for making presentable and informative statistical graphics which makes it ideal for business and marketing analysis.
- This library helps you learn and explore about data.

Installation and Execution

- We can use the following line of script to install Seaborn using pip −
  - pip3 install seaborn
- We can import Seaborn to our Python script using the following lines of script −
  - import seaborn as sns
- After installing Seaborn, you can import it into your Python script

Keras

- Keras is an high level neural network library that creates deep learning models.
- It runs on top of TensorFlow, CNTK, or Theano.
- It provides a simple and intuitive API for building and training deep learning models, making it an excellent choice for beginners and researchers.
- Keras is one of the popular library as it allows for easy and fast prototyping

Installation and Execution

For CPU platform on Windows operating system, use the following to install Keras using pip −

- pip3 install keras
- To import TensorFlow use the following −
  - import keras
- After installing Keras, you can import it into your Python script

Example - Using CIFAR-10 dataset

In the example below, we are importing CIFAR-10 dataset from Keras and printing the shape of training data and test data −

```
import keras
(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

Output

The above example code will produce the following result −

```
(50000, 32, 32, 3)
(10000, 32, 32, 3)
(50000, 1)
(10000, 1)
```