Experiment 1

Aim: Study of Machine Learning Libraries and tools (Python library, TensorFlow, keras,...)

Objective: Ability to understand the different libraries and their function used for data preprocessing and model development.

Theory:

Python is one of the most popular and widely used programming languages and has replaced many programming languages in the industry.

There are a lot of reasons why Python is popular among developers and one of them is that it has an amazingly large collection of libraries that users can work with. It contains a numerous number of algorithms for implementing standard machine learning and data mining tasks like reducing dimensionality, classification, regression, clustering, and model selection.

Numpy

Numpy is considered as one of the most popular machine learning library in Python.

TensorFlow and other libraries uses Numpy internally for performing multiple operations on Tensors. Array interface is the best and the most important feature of Numpy.

Features Of Numpy

1. **Interactive:** Numpy is very interactive and easy to use.
2. **Mathematics:** Makes complex mathematical implementations very simple.
3. **Intuitive:** Makes coding real easy and grasping the concepts is easy.
4. **Lot of Interaction:** Widely used, hence a lot of open source contribution.

This interface can be utilized for expressing images, sound waves, and other binary raw streams as an array of real numbers in N-dimensional.

For implementing this library for machine learning having knowledge of Numpy is important for full stack developers.

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.

72-Mokshad Sankhe

- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

Pandas
What Is Pandas?

Pandas is a machine learning library in Python that provides data structures of high-level and a wide variety of tools for analysis. One of the great feature of this library is the ability to translate complex operations with data using one or two commands. Pandas have so many inbuilt methods for grouping, combining data, and filtering, as well as time-series functionality.

Features Of Pandas

Pandas make sure that the entire process of manipulating data will be easier. Support for operations such as Re-indexing, Iteration, Sorting, Aggregations, Concatenations and Visualizations are among the feature highlights of Pandas.

Applications of Pandas?

Currently, there are fewer releases of pandas library which includes hundred of new features, bug fixes, enhancements, and changes in API. The improvements in pandas regards its ability to group and sort data, select best suited output for the apply method, and provides support for performing custom types operations.

Data Analysis among everything else takes the highlight when it comes to usage of Pandas. But, Pandas when used with other libraries and tools ensure high functionality and good amount of flexibility.

Keras
What Is Keras?
Keras is considered as one of the coolest machine learning libraries in Python. It provides an easier mechanism to express neural networks. Keras also provides some of the best utilities for compiling models, processing data-sets, visualization of graphs, and much more.

In the backend, Keras uses either Theano or TensorFlow internally. Some of the most popular neural networks like CNTK can also be used. Keras is comparatively slow when we compare it with other machine learning libraries. Because it creates a computational graph by using

back-end infrastructure and then makes use of it to perform operations. All the models in Keras are portable.'

Features Of Keras

- It runs smoothly on both CPU and GPU.
- Keras supports almost all the models of a neural network – fully connected, convolutional, pooling, recurrent, embedding, etc. Furthermore, these models can be combined to build more complex models.

- Keras, being modular in nature, is incredibly expressive, flexible, and apt for innovative research.
- Keras is a completely Python-based framework, which makes it easy to debug and explore.

Where are we using Keras?

You are already constantly interacting with features built with Keras — it is in use at Netflix, Uber, Yelp, Instacart, Zocdoc, Square, and many others. It is especially popular among startups that place deep learning at the core of their products.

Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers and a host of tools to make working with image and text data easier.

Plus, it provides many pre-processed data-sets and pre-trained models like MNIST, VGG, Inception, SqueezeNet, ResNet etc.

Keras is also a favorite among deep learning researchers

TensorFlow

What Is TensorFlow?

If you are currently working on a machine learning project in Python, then you may have heard about this popular open source library known as TensorFlow.

This library was developed by Google in collaboration with Brain Team. TensorFlow is a part of almost every Google application for machine learning.

TensorFlow works like a computational library for writing new algorithms that involve a large number of tensor operations, since neural networks can be easily expressed as computational graphs they can be implemented using TensorFlow as a series of operations on Tensors. Plus, tensors are N-dimensional matrices which represent your data.

Features of TensorFlow

TensorFlow is optimized for speed, it makes use of techniques like XLA for quick linear algebra operations.

1. Responsive Construct

With TensorFlow, we can easily visualize each and every part of the graph which is not an option while using *Numpy* or *SciKit*.

2. Flexible

One of the very important Tensorflow Features is that it is flexible in its operability, meaning it has modularity and the parts of it which you want to make standalone, it offers you that option.

3. Easily Trainable

It is easily trainable on CPU as well as *GPU* for distributed computing.

72-Mokshad Sankhe

4. Parallel Neural Network Training

TensorFlow offers pipelining in the sense that you can train multiple *neural networks*and multiple GPUs which makes the models very efficient on large-scale systems.

5. Large Community

Needless to say, if it has been developed by Google, there already is a large team of software engineers who work on stability improvements continuously.

6. Open Source

The best thing about this machine learning library is that it is open source so anyone can use it as long as they have internet connectivity.

Uses of TensorFlow?

You are using TensorFlow daily but indirectly with applications like Google Voice Search or Google Photos. These are the applications of TensorFlow.

All the libraries created in TensorFlow are written in C and C++. However, it has a complicated front-end for Python. Your Python code will get compiled and then executed on TensorFlow distributed execution engine built using C and C++.

The number of applications of TensorFlow is literally unlimited and that is the beauty of TensorFlow .

Implementation:

Matplotlib

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python.

Program:

```
import matplotlib.pyplot as plt
%matplotlib inline
x = [1, 2, 3, 4, 5, 6, 7]
y = [50, 51, 52, 48, 47, 49, 46]
plt.xlabel('day')
plt.ylabel("Temprature")
plt.title('Weather')
plt.plot(x, y, color='green', linewidth=5, linestyle='dashdot')
```

Output:
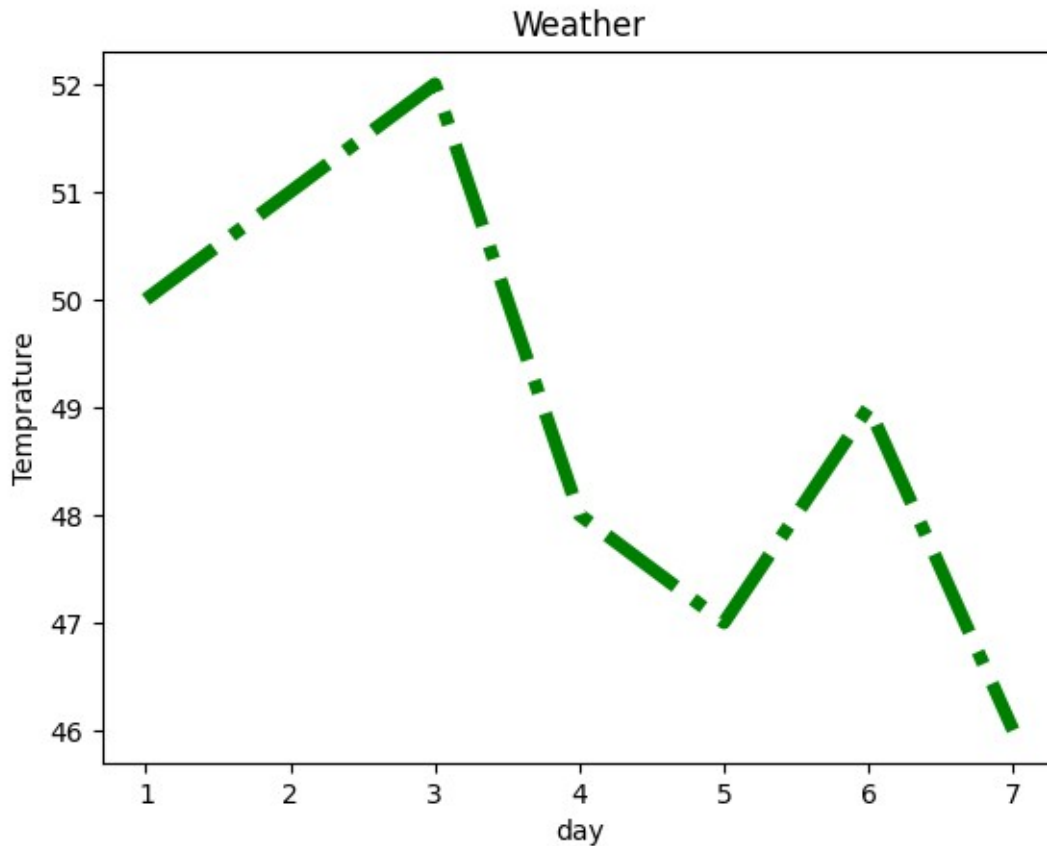
PANDAS

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython.

Program:

```
import pandas as pd
import numpy as np
import sys
fruits = np.array(['apple', 'orange', 'mango', 'pear'])
series2 = pd.Series (fruits)
print(series2)
```

Output:
```
0    apple
1    orange
2    mango
3    pear
dtype: object
```

NUMPY

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:A powerful N-dimensional array object Sophisticated (broadcasting) functions Tools for integrating C/C++ and Fortran code Useful linear algebra, Fourier transform, and random number capabilities

Program:
```
import numpy as np
import time
# converting list and set to array
set = {1, 2, 3, 4, 5}
list = [1, 2, 3, 4, 5]
arr = np.array(list)
arr2 = np.array(set)
print(arr)
print(arr2)
arr = [[0,2,4,6,8], [1,3,5,7,9]] #no of column is 5 in both
arr = np.array(arr)
print(arr)
print(arr.shape)
print(arr[0])
print(arr[1])
arr = np.zeros((3,7), dtype=int)
print(arr)
arr  = np.arange(0,10,2)
arr1 = np.arange(0, 20, 2).reshape(5, 2)
arr = np.arange(4)
print(arr)
arr =  np.linspace (0,2,5, dtype=int)
print(arr)
```

Output:

```
[1 2 3 4 5]
{1, 2, 3, 4, 5}
[[0 2 4 6 8]
 [1 3 5 7 9]]
(2, 5)
[0 2 4 6 8]
[1 3 5 7 9]
[[0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]]
[0 1 2 3]
[0 0 1 1 2]
```

TensorFlow:

TensorFlow is an open-source machine learning library developed by Google Brain Team. It provides a comprehensive ecosystem of tools, libraries, and community resources to build and deploy machine learning models effectively.

TensorFlow is primarily used for developing deep learning models, including neural networks for various tasks such as image classification, natural language processing, and reinforcement learning.

TensorFlow offers high-level APIs like Keras for easy model building and training, as well as lower-level APIs for advanced customization and optimization.

Example usage: Building and training convolutional neural networks (CNNs) for image recognition tasks, sequence models for natural language processing, and generative adversarial networks (GANs) for image generation.

Scikit-learn:

Scikit-learn is a simple and efficient tool for data mining and data analysis built on NumPy, SciPy, and matplotlib. It provides a wide range of supervised and unsupervised learning algorithms for classification, regression, clustering, dimensionality reduction, and model selection.

Scikit-learn is designed to be easy to use and accessible, making it an excellent choice for beginners and experienced practitioners alike.

It offers consistent and intuitive APIs, extensive documentation, and a rich set of features for preprocessing data, evaluating models, and tuning hyperparameters.

Example usage: Implementing machine learning algorithms such as support vector machines (SVM), decision trees, random forests, k-nearest neighbors (KNN), and Gaussian mixture models (GMM) for various predictive modeling tasks.

Keras:

Keras is an open-source deep learning framework written in Python. It is designed to be user-friendly, modular, and extensible, allowing rapid prototyping and experimentation with deep neural networks. Keras provides a high-level neural networks API that can run on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). However, it has been integrated into TensorFlow as its official high-level API since TensorFlow 2.0.

With Keras, users can easily define and train deep learning models, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more complex architectures such as autoencoders and generative adversarial networks (GANs).

Example usage: Building and training deep learning models for image classification, object detection, sequence prediction, and reinforcement learning tasks. Keras offers a simple and intuitive interface for defining layers, specifying loss functions, and configuring optimization algorithms.

Conclusion:

In this experiment, we explored various machine learning libraries and tools in Python, including NumPy, Matplotlib, Pandas, Keras, TensorFlow, and Scikit-learn. These libraries offer essential features for data preprocessing, visualization, model development, and deployment. NumPy provides efficient array operations, Matplotlib facilitates visualization, Pandas simplifies data analysis, Keras offers a high-level API for neural network development, TensorFlow enables scalable machine learning computations, and Scikit-learn provides a wide range of algorithms for data mining tasks. Understanding and utilizing these libraries are crucial for developing effective machine learning models and gaining insights from data.