Roll No.: A56 Class : TE - A Name: Shubham Mangaonkar
Date of Experiment: 08/02/24 Date of Submission: 16/02/24

Experiment No.02

Aim: To study Machine Learning libraries and tools like tensorflow, keras, etc.

Python Libraries:

Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries. Adding the right mix of libraries to your development environment is crucial. The following packages and libraries are vital for most AI developers. All of them are freely available as open source distros.

Pandas:

Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.

In [2]:
```
# Load our necessary libraries
import pandas as pd
import numpy as np
```

In [3]:
```
# Create data into CSV (that we'll import later on)
# Let's say the data is for a veterinarian keeping tabs on his clients.

raw_data = {'pet_name': ['Woof', 'Chester', 'Rex', 'Mystery', 'Pumpkin'],
        'pet_last_name': ['Smith', 'Kim', "", 'Taylor', ""],
        'good_pet_score': [96, 34, 89, 92, 79],
        'type': ['dog', 'cat', 'mini-dinosaur', "unknown", "bird"],
        'amount_owed': ["5000", "9,000", 570, 622, 190]}
df = pd.DataFrame(raw_data, columns = ['pet_name', 'pet_last_name', 'good_pet_score', 'type', 'amount_owed'])
df
```

Out[3]:

	pet_name	pet_last_name	good_pet_score	type	amount_owed
0	Woof	Smith	96	dog	5000
1	Chester	Kim	34	cat	9,000
2	Rex		89	mini-dinosaur	570
3	Mystery	Taylor	92	unknown	622
4	Pumpkin		79	bird	190

Features: The core feature of Pandas is its variety of data structures, which let users perform an assortment of analysis operations.

- Pandas has a variety of modules for data manipulation, including reshape, join, merge, and pivot.
- Pandas has data visualization capabilities.
- Users can perform mathematical operations including calculus and statistics without calling on outside libraries.
- It has modules that help you work around missing data.

Best for: Data analysis.

Scikit-learn:

Scikit-learn is a Python library for implementing machine learning algorithms.

Background: A developer named David Cournapeau originally released scikit-learn as a student in 2007. The open source community quickly adopted it and has updated it numerous times over the years.

```
In [3]: from sklearn.datasets import load_iris
        iris = load_iris()

        X = iris.data
        y = iris.target

        feature_names = iris.feature_names
        target_names = iris.target_names

        print("Feature names:", feature_names)
        print("Target names:", target_names)

        print("\nType of X is:", type(X))

        print("\nFirst 5 rows of X:\n", X[:5])

Feature names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
Target names: ['setosa' 'versicolor' 'virginica']

Type of X is: <class 'numpy.ndarray'>

First 5 rows of X:
 [[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
```

Roll No.: A56 Class : TE - A Name: Shubham Mangaonkar
Date of Experiment: 08/02/24 Date of Submission: 14/02/24

Features: The packages in Scikit-learn focus on modeling data.

- Scikit-learn includes every core machine learning algorithm, among them vector machines, random forests, gradient boosting, k-means clustering, and DBSCAN.
- It was designed to work seamlessly with NumPy and SciPy (both described below) for data cleaning, preparation, and calculation.
- It has modules for loading data as well as splitting it into training and test sets.
- It supports feature extraction for text and image data.

Best for: Scikit-learn is a must-have for anybody working in machine learning. It is considered one of the best libraries available if you need to implement algorithms for classification, regression, clustering, model selection, and more.

NumPy:

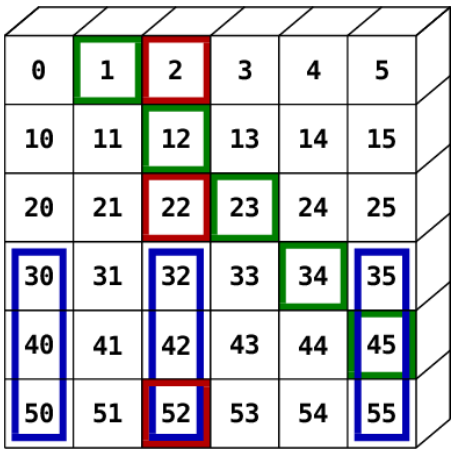
NumPy is a Python package for working with arrays, or large collections of homogenous data. You can think of an array like a spreadsheet, where numbers are stored in columns and rows.

Background: Python wasn't originally intended for numerical computation when it was launched in 1991. Still, its ease of use caught the scientific community's attention early on. Over the years, the open source community developed a succession of packages for numerical computing. In 2005, developer Travis Oliphant combined over a decade's worth of open source developments into a single library for numerical computation, which he called NumPy.

```
>>> a[(0,1,2,3,4), (1,2,3,4,5)]
array([1, 12, 23, 34, 45])

>>> a[3:, [0,2,5]]
array([[30, 32, 35],
       [40, 42, 45],
       [50, 52, 55]])

>>> mask = np.array([1,0,1,0,0,1], dtype=bool)
>>> a[mask, 2]
array([2, 22, 52])
```



Features: The core feature of NumPy is support for arrays, which allows you to quickly process and manipulate large collections of data.

- Arrays in NumPy can be n-dimensional. This means the data can be a single column of numbers, or many columns and rows of numbers.
- NumPy has modules for performing some linear algebra functions.
- It also has modules for graphing and plotting numerical arrays.
- Data in NumPy arrays is homogenous, which means it must all be defined as the same type (numbers, strings, Boolean values, etc.). This means data gets processed efficiently.

Best for: Manipulating and processing data for more advanced data science or machine learning operations. If you are crunching numbers, you need NumPy.

SciPy:

SciPy is a Python library for scientific computing. It contains packages and modules for performing calculations that help scientists conduct or analyze experiments.

Background: In the late 1990s and early 2000s, the Python open source community began working on a collection of tools to meet the needs of the scientific community. In 2001, they released these tools as SciPy. The community remains active and is always updating and adding new features.

```
In [1]: # Importing the sub-packge optimize as ot
        import scipy.optimize as ot

In [2]: # defining the Objectve function that neet to be minimized
        def Objective_Fun(x):

            return 2*x**2+5*x-4

In [3]: # Importing the method minimize_scalar
        #and computing the minimum value of objective function

        result = ot.minimize_scalar(Objective_Fun)

In [4]: # checking the minimum value of Objective function
        print(result)

             fun: -7.125
         message: '\nOptimization terminated successfully;\nThe returned value
        08 )'
            nfev: 9
             nit: 4
         success: True
               x: -1.2500000185100002
```

Features: SciPy's packages comprise a complete toolkit of mathematical techniques from calculus, linear algebra, statistics, probabilities, and more.

- Some of its most popular packages for data scientists are for interpolation, K-means testing, numerical integration, Fourier transforms, orthogonal distance regression, and optimization.
- SciPy also includes packages for image processing and signal processing.
- The Weave feature allows users to write code in C/C++ within Python.

Best for: SciPy is a data scientist's best friend.

TensorFlow

TensorFlow is an end-to-end open source library for developing, training, and deploying deep learning models. TensorFlow was originally released in 2015 by Google Brain. Originally, its front end wasn't user friendly, and it had redundant APIs that made building and implementing models cumbersome. Many of these issues have been resolved over time with updates, as well as by integrating Keras (see below) as the default front end.

```
import tensorflow_cloud as tfc

tfc.run(
    requirements_txt="requirements.txt",
    chief_config=tfc.MachineConfig(
        cpu_cores=8,
        memory=30,
        accelerator_type=tfc.AcceleratorType.NVIDIA_TESLA_T4,
        accelerator_count=2,
    ),
    docker_image_bucket_name=GCP_BUCKET,
)
```

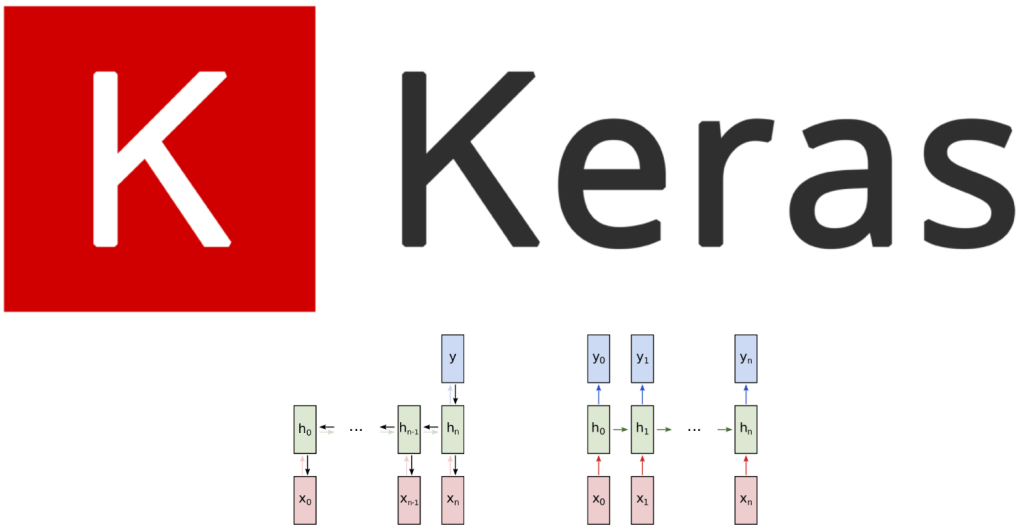
Features: TensorFlow has numerous packages for building deep learning models and scaling them for commercial deployment.

- TensorFlow users can call upon the hundreds of pre-trained models in the Dev Hub and Model Garden. The Dev Hub contains plug-and-play models while the Model Garden is intended for more advanced users who are comfortable making customizations.
- It is efficient in its use of memory, making it possible to train multiple neural networks in parallel.
- TensorFlow applications can run on a wide variety of hardware systems, including CPUs, GPUs, TPUs, and more.
- TensorFlow Lite is optimized for mobile and embedded machine learning models.
- Users can freely upload and share their machine learning experiments on Tensorboard.dev.

Best for: Building production-ready deep learning models at scale.

Keras

Keras is a beginner-friendly toolkit for working with neural networks. It is the front-end interface for TensorFlow. Google engineer Francois Choillet released Keras in 2015 to act as an API for a number of deep learning libraries. As of 2020, Keras is exclusive to TensorFlow.



Features: Keras handles the high level tasks of building neural networks in TensorFlow, and as such contains fundamental modules like activation functions, layers, optimizers, and more.

- Keras supports vanilla neural networks, convolutional neural networks, and recurrent neural networks as well as utility layers including batch normalization, dropout, and pooling.
- It is designed to simplify coding deep neural networks.

Best for: Developing deep learning networks.

Conclusion:

Thus ,we have successfully explored and studied about various machine learning libraries like Scikit-learn, Tensorflow and Keras.