Assignment No. 1
Topic - Python Libraries for Machine Learning

Aug 28, 2021

GroupID: DM21G16

Group Members: 2018BTECS00050 Rushikesh Shelke

2018BTECS00064 Saurabh Hirugade

Aim: To study and implement Machine Learning libraries in python.

Introduction:

What are python libraries?

A Python library is a reusable chunk of code that you may want to include in your programs/ projects.
They are particularly useful for accessing the pre-written frequently used codes, instead of writing them from scratch every single time.
Python libraries play a vital role in developing machine learning, data science, data visualization, image and data manipulation applications and more.

Python Library is basically a collection of modules. Let us understand what is module through example,
Wheel has already been invented, So the person who invented the car didn't waste his time in re-inventing the wheel again. Here, the car is an invention which has an imported wheel. so, the wheel is a module (can be used in other inventions as it is).

Why is Python so popular for Machine Learning?

1. The availability of libraries and open source tools make it an ideal choice for developing ML models.
2. Py has solutions for every existing problem in ML.
3. Syntax of python and imports are so intuitive, people with non-cs background can also understand it easily.
4. Python is a highly scalable language and is also much faster than other languages such as R, Stata, and Matlab. Its scalability further enhances its flexibility quotient, which is extremely useful in problem-solving and app development.

Theory/Documentation:

Top Python Libraries:

1. NumPy

NumPy is a well known general-purpose array-processing package. An extensive collection of high complexity mathematical functions make NumPy powerful enough to process large multi-dimensional arrays and matrices. NumPy is very useful for handling linear algebra, Fourier transforms, and random numbers. Other libraries like TensorFlow uses NumPy at the backend for manipulating tensors.

With NumPy, you can define arbitrary data types and easily integrate with most databases. NumPy can also serve as an efficient multi-dimensional container for any generic data that is in any datatype. The key features of NumPy include

powerful N-dimensional array object, broadcasting functions, and out-of-box tools to integrate C/C++ and Fortran code.

Advantages:

1. Intuitive and interactive.
2. Offers Fourier transforms, random number capabilities, and other tools for integrating computing languages like C/C++ and Fortran.
3. Versatility – other ML libraries like scikit-learn and TensorFlow use NumPy arrays as input; data manipulation packages like Pandas use NumPy under the hood.
4. Has terrific open-source community support/contributions.
5. Simplifies complex mathematical implementations.

Disadvantages:

1. Can be overkill – do not use when you can get away with Python Lists, instead.

2. SciPy

The SciPy library offers modules for linear algebra, image optimization, integration interpolation, special functions, Fast Fourier transform, signal and image processing, Ordinary Differential Equation (ODE) solving, and other computational tasks in science and analytics.

The underlying data structure used by SciPy is a multi-dimensional array provided by the NumPy module. SciPy depends on NumPy for the array manipulation subroutines. The SciPy library was built to work with NumPy arrays along with providing user-friendly and efficient numerical functions.

Advantages:

1. Great for image manipulation.
2. Provides easy handling of mathematical operations.
3. Offers efficient numerical routines, including numerical integration and optimization.
4. Supports signal processing.

Disadvantages:

1. There is both a stack and a library named SciPy. The library is part of the stack. Beginners who don't know the difference may become confused.

3. Scikits-learn

In 2007, David Cournapeau developed the Scikit-learn library as part of the Google Summer of Code project. In 2010 INRIA involved and did the public release in January 2010. Scikit-learn was built on top of two Python libraries – NumPy and SciPy and has become the most popular Python machine learning library for developing machine learning algorithms.

Scikit-learn has a wide range of supervised and unsupervised learning algorithms that works on a consistent interface in Python. The library can also be used for data-mining and data analysis. The main machine learning functions that the Scikit-learn library can handle are classification, regression, clustering, dimensionality reduction, model selection, and preprocessing.

Advantages:

1. Simple, easy to use, and effective.
2. In rapid development, and constantly being improved.
3. Wide range of algorithms, including clustering, factor analysis, principal component analysis, and more.
4. Can extract data from images and text.
5. Can be used for NLP.

Disadvantages:

1. This library is especially suited for supervised learning, and not very suited to unsupervised learning applications like Deep Learning.

4. Theano

Theano is a python machine learning library that can act as an optimizing compiler for evaluating and manipulating mathematical expressions and matrix calculations. Built on NumPy, Theano exhibits a tight integration with NumPy and has a very similar interface. Theano can work on Graphics Processing Unit (GPU) and CPU.

Working on GPU architecture yields faster results. Theano can perform data-intensive computations up to 140x faster on GPU than on a CPU. Theano can automatically avoid errors and bugs when dealing with logarithmic and exponential functions. Theano has built-in tools for unit-testing and validation, thereby avoiding bugs and problems.

5. Tensorflow

TensorFlow was developed for Google's internal use by the Google Brain team. Its first release came in November 2015 under Apache License 2.0. TensorFlow is a popular computational framework for creating machine learning models. TensorFlow supports a variety of different toolkits for constructing models at varying levels of abstraction.

TensorFlow exposes very stable Python and C++ APIs. It can expose backward compatible APIs for other languages too, but they might be unstable. TensorFlow has a flexible architecture with which it can run on a variety of computational platforms CPUs, GPUs, and TPUs. TPU stands for Tensor processing unit, a hardware chip built around TensorFlow for machine learning and artificial intelligence.

Advantages:

1. Supports reinforcement learning and other algorithms.
2. Provides computational graph abstraction.
3. Offers a very large community.
4. Provides TensorBoard, which is a tool for visualizing ML models directly in the browser.
5. Production ready.
6. Can be deployed on multiple CPUs and GPUs.

Disadvantages:

1. Runs dramatically slower than other frameworks utilizing CPUs/GPUs.
2. Steep learning curve compared to PyTorch.
3. Computational graphs can be slow.
4. Not commercially supported.
5. Not very toolable.

6. Keras

Keras has over 200,000 users as of November 2017. Keras is an open-source library used for neural networks and machine learning. Keras can run on top of TensorFlow, Theano, Microsoft Cognitive Toolkit, R, or PlaidML. Keras also can run efficiently on CPU and GPU.

Keras works with neural-network building blocks like layers, objectives, activation functions, and optimizers. Keras also has a bunch of features to work on images and text images that comes handy when writing Deep Neural Network code.

Apart from the standard neural network, Keras supports convolutional and recurrent neural networks.

Advantages:

1. Great for experimentation and quick prototyping.
2. Portable.
3. Offers easy expression of neural networks.
4. Great for use in modeling and visualization.

Disadvantages:

1. Slow, since it needs to create a computational graph before it can perform operations.

7. PyTorch

PyTorch has a range of tools and libraries that support computer vision, machine learning, and natural language processing. The PyTorch library is open-source and is based on the Torch library. The most significant advantage of PyTorch library is it's ease of learning and using.

PyTorch can smoothly integrate with the python data science stack, including NumPy. You will hardly make out a difference between NumPy and PyTorch. PyTorch also allows developers to perform computations on Tensors. PyTorch has a robust framework to build computational graphs on the go and even change them in runtime. Other advantages of PyTorch include multi GPU support, simplified preprocessors, and custom data loaders.

Advantages:

1. Contains tools and libraries that support Computer Vision, NLP , Deep Learning, and many other ML programs.
2. Developers can perform computations on Tensors with GPU acceleration.
3. Helps in creating computational graphs.
4. Modeling process is simple and transparent.
5. The default "define-by-run" mode is more like traditional programming.
6. Uses common debugging tools such as pdb, ipdb or PyCharm debugger.
7. Uses a lot of pre-trained models and modular parts that are easy to combine.

Disadvantages:

1. Because PyTorch is relatively new, there are comparatively fewer online resources to be found. This makes it harder to learn from scratch, although it is intuitive.
2. PyTorch is not widely considered to be production-ready compared to Google's TensorFlow, which is more scalable.

8. Pandas

Pandas are turning up to be the most popular Python library that is used for data analysis with support for fast, flexible, and expressive data structures designed to work on both "relational" or "labeled" data. Pandas today is an inevitable library for solving practical, real-world data analysis in Python. Pandas is highly stable, providing highly optimized performance. The backend code is purely written in C or Python.

The two main types of data structures used by pandas are :

1. Series (1-dimensional)
2. DataFrame (2-dimensional)

These two put together can handle a vast majority of data requirements and use cases from most sectors like science,

statistics, social, finance, and of course, analytics and other areas of engineering.

Advantages:

1. Expressive, fast, and flexible data structures.
2. Supports aggregations, concatenations, iteration, re-indexing, and visualizations operations.
3. Very flexible usage in conjunction with other Python libraries.
4. Intuitive data manipulation using minimal commands.
5. Supports a wide range of commercial and academic domains.
6. Optimized for performance.

Disadvantages:

1. It is built on matplotlib, meaning a novice programmer has to be familiar with both libraries in order to know which one would be best suited to solve their problem.
2. Less suitable for n-dimensional arrays and statistical modeling. Use NumP, SciPy or SciKit Learn instead.

9. Matplotlib

Matplotlib is a data visualization library that is used for 2D plotting to produce publication-quality image plots and figures in a variety of formats. The library helps to generate histograms, plots, error charts, scatter plots, bar charts with just a few lines of code.

It provides a MATLAB-like interface and is exceptionally user-friendly. It works by using standard GUI toolkits like GTK+, wxPython, Tkinter, or Qt to provide an object-oriented API that helps programmers to embed graphs and plots into their applications.

Advantages:

1. Flexible usage: supports both Python and IPython shells, Python scripts, Jupyter Notebook, web application servers and many GUI toolkits (GTK+, Tkinter, Qt, and wxPython).
2. Optionally provides a MATLAB-like interface for simple plotting.
3. The object-oriented interface gives complete control of axes properties, font properties, line styles, etc.
4. Compatible with several graphics backends and operating systems.
5. Matplotlib is frequently incorporated in other libraries, such as Pandas.

Disadvantages:

1. Because Matplotlib has two different interfaces (object-oriented vs MATLAB-like), a novice developer can become confused.
2. Matplotlib is a visualization library, not a data analysis library. For data analysis, you'll need to combine it with other libraries, like Pandas.

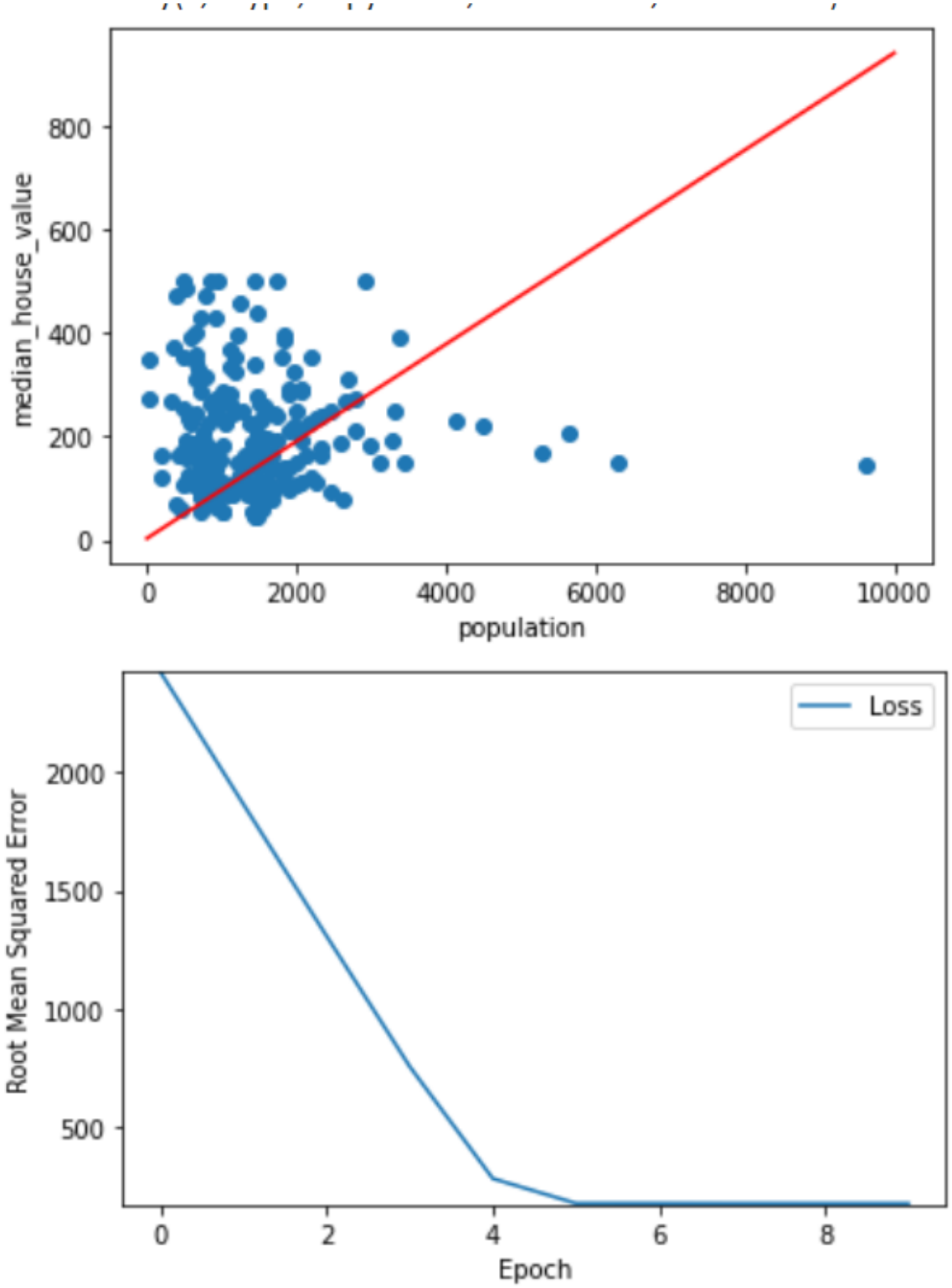
Actual Experimentation:

We wrote a program to predict the prices of houses from given dataset. Libraries used: *(Code file is separately included)*

1. Pandas

2. Tensorflow
3. Matplotlib

Data visualization output:



Output of program:

```
predict_house_values(20, my_feature, my_label)
```

```
feature   label        predicted
 value    value        value
          in thousand$   in thousand$
-------------------------------------
 1286     53           124
 1867     92           178
 2191     69           208
 1052     62           102
 1647     80           158
 2312    295           220
 1604    500           153
 1066    342           103
  338    118            35
 1604    128           153
 1200    187           116
  292     80            31
 2014    112           192
 1817     95           173
 1328     69           128
 2133     90           203
 1929     54           184
  966     68            94
 1143     71           110
  959     73            93
```

Conclusion:

1. We studied various libraries that are present in python that have applications in Machine Learning.

2. Python is the go-to language when it comes to data science and machine learning and there are multiple reasons to choose python for data science.
3. We implemented some of the well known libraries in python and experienced it's functionality.

References:

1. https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/
2. https://www.upgrad.com/blog/python-for-data-science/
3. https://www.mdpi.com/2078-2489/11/4/193
4. https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c