Hands-on Lab Description

CS-ML-00199 – Python Machine Learning Tutorial-Basic Concepts

Category:

```
CS-ML: Machine Learning
```

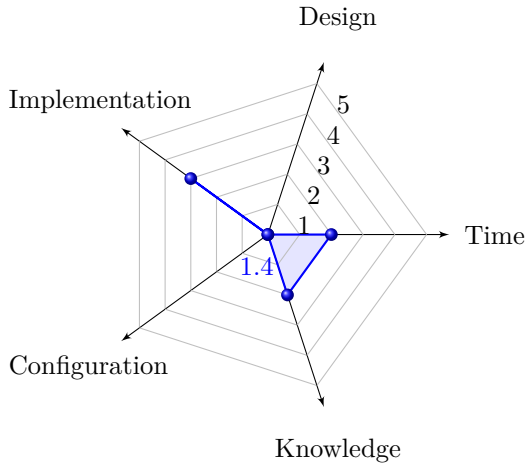
Objectives:

```
Understand basic concepts of Python
Understand why Python is a good programming language to solution data science
    problems.
```

Estimated Lab Duration:

```
Expert: 30 minutes
Novice: 120 minutes
```

Difficulty Diagram:



Difficulty Table.

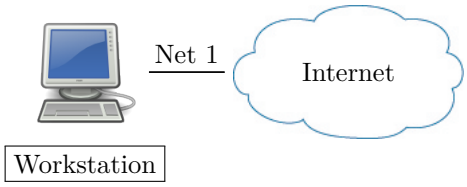
Measurements	Values (0-5)
Time	2
Design	0
Implementation	3
Configuration	0
Knowledge	2
Score (Average)	1.4

Required OS:

```
Linux: Ubuntu 18.04 LTS
```

Lab Running Environment:

```
VirtualBox https://www.virtualbox.org/ (Reference Labs: CS-SYS-00101)
```




```
1 Server: Linux (Ubuntu 18.04 LTS)
2 Network Setup: connected to the Internet
```

Lab Preparations:

```
Python and Anaconda software packages installed on the Linux VM. Reference Lab:
    CS-ML-00001
```

Introduction

Data science is a way to try and discover hidden patterns in raw data. It is used to extract knowledge or insights from data in various forms, either structured or unstructured. To achieve this goal, it makes use of several algorithms, Machine Learning (ML) principles, and scientific methods.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics, from which its inventer Guido van Rossum named it after the comedy group Monty Python. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance, which make it a de facto choice for data scientists. Some of the libraries well known in the data science community – Pandas, StatsModels, NumPy, SciPy, Keras, Scikit-Learn, etc., and some 72,000 of them in the Python Package Index (PyPI) and still growing constantly.

Here, we enumerate several essential python libraries for data scientists.

Python Pandas

Python Pandas is considered the most important and widely used python library for data science. From importing data from spreadsheets to processing sets for time-series analysis, Pandas is used for everything. Pandas pretty much convert one data form to another on your fingertips. Hence, Pandas powerful data frames can perform both, basic cleanup and advance data manipulations.

Behind Python's data science success story, one of the earliest libraries is Numpy (Numerical Python), on which Pandas is built. NumPy's functions exposure is used in Pandas for advanced analysis. For more specialization, one can use Scipy which is scientifically equivalent to Numpy, offering tools and techniques for scientific data analysis.

NumPy

NumPy facilitates easy and efficient numeric computation. It lets you deal with large, multi-dimensional arrays and matrices. To act on these, it also gives us various high-level mathematical functions.

SciPy

SciPy will give you all the tools you need for scientific and technical computing. It has modules for optimization, linear algebra, integration, interpolation, special functions, Fast Fourier Transfer (FFT), signal and image processing, Ordinary Differential Equation (ODE) solvers, and other tasks.

Matplotlib

Python also provides powerful visualization libraries – Matplotlib. It can be used in all kinds of GUI toolkits such as python scripts, web applications as well as shell, etc. With an object-oriented API, it lets you embed plots into applications. For this, it uses GUI toolkits like Tkinter, Qt, GTK+, and wxPython. You have the opportunity to use different types of plots and work with multiple plots.

Scikit – Learn and Pybrain

Scikit – Learn & Pybrain, one of the attractions of python where you implement machine learning. With the support of simple and efficient tools in this library which can be used for data analysis and data mining.

Various algorithms have their back, such as classification, logistic regression, clustering, time series, etc., and it is usually used alongside NumPy and SciPy.

- Support Vector Machines
- Random forests
- Gradient boosting
- K-means
- DBSCAN

Seaborn

Seaborn is a visualization library for Python and is based on matplotlib. It allows data scientists to perform data visualization in a statistical manner with a high-level interface that results in attractive graphics.

TensorFlow

TensorFlow is the most popular tool for Machine Learning in Python. It was developed specifically for carrying out deep learning operations. The basic data structure of TensorFlow ecosystem are the tensors. As a matter of fact, the name of TensorFlow is derived from these tensors. TensorFlow is continuously evolving owing to an open-source community who have made it a pioneering toolkit for machine learning operations. It provides support for CPUs, GPUs as well as TPUs. Due to this, it provides lightning speed execution speed for various machine learning algorithms.

TensorFlow has numerous applications. This is mainly because of its high processing capability. It is used for the development of speech recognition product, recommendation systems, Generative Adversarial Networks, etc. TensorFlow is basically the standardized tool for performing Deep Learning operations.

Install and Setup Python

Anaconda is a python distribution for data science and machine learning. It is free and open-source and makes managing and deploying packages simple. It has more than 1000 data science packages and the Conda package. Other tools it comes with are core Python, IPython, among others. For more details on how to install anaconda and setup python running environment, please refer to the lab 'Setup Machine Learning Running Environment on Linux' (CS-ML-00001).

Comparison between Python 2 and Python 3

There are two major versions of python version 2 and version 3. Python 2 implemented technical details of Python Enhancement Proposal (PEP). Python 2.7 (last version in 2.x) is no longer under development and it was discontinued in 2020.

On December 2008, Python released version 3.0. This version was mainly released to fix problems which exist in Python 2. The nature of these change is such that Python 3 was incompatible with Python 2. It is backward incompatible Some features of Python 3 have been backported to Python 2.x versions to make the migration process easy in Python 3.

As a result, for any organization who was using Python 2.x version, migrating their project to 3.x needed lots of changes. These changes not only relate to projects and applications but also all the libraries that form part of the Python ecosystem.

Here, are prime reasons for using Python 3.x versions:

- Python 3 supports modern techniques like AI, machine learning, and data science
- Python 3 is supported by a large Python developer's community.
- Its easier to learn Python language compared to earlier versions.

- Offers Powerful toolkit and libraries
- Mixable with other languages

In Table CS-ML-00199.2, it shows a comparative features study between Python version 2 and version 3:

Table CS-ML-00199.2: Comparisons between Python 2 and Python 3.

Basis of comparison	Python 3	Python 2
Release Date	2008	2000
Function print	print ("hello")	print "hello"
Division of Integers	Whenever two integers are divided, you get a float value	When two integers are divided, you always provide integer value.
Unicode	In Python 3, default storing of strings is Unicode.	To store Unicode string value, you require to define them with "u".
Syntax	The syntax is simpler and easily understandable.	The syntax of Python 2 was comparatively difficult to understand.
Rules of ordering Comparisons	In this version, Rules of ordering comparisons have been simplified.	Rules of ordering comparison are very complex.
Iteration	The new Range() function introduced to perform iterations.	In Python 2, the xrange() is used for iterations.
Exceptions	It should be enclosed in parenthesis.	It should be enclosed in notations.
Leak of variables	The value of variables never changes.	The value of the global variable will change while using it inside for-loop.
Backward compatibility	Not difficult to port python 2 to python 3 but it is never reliable.	Python version 3 is not backwardly compatible with Python 2.
Library	Many recent developers are creating libraries which you can only use with Python 3.	Many older libraries created for Python 2 is not forward-compatible.

Summary

Programmers often fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python Basics

The Python programming language is an object-oriented language, which means that it can model real-world entities. It is also dynamically-typed because it carries out type-checking at run-time. It does so to make sure

that the type of construct matches what we expect it to be. The distinctive feature of Python is that it is an interpreted language. The Python IDLE (Integrated Development Environment) executes instructions one line at a time. Thus, it also can be used as a calculator.

Python Constructs

“nobreak

Functions

A function in Python is a collection of statements grouped under a name. You can use it whenever you want to execute all those statements at a time. You can call it wherever you want and as many times as you want in a program. A function may return a value.

Classes

Python is an object-oriented language, and thus, it supports classes and objects. A class is an abstract data type. In other words, it is a blueprint for an object of a certain kind. It holds no values. An object is a real-world entity and an instance of a class.

Modules

A Python module is a collection of related classes and functions. It has modules for mathematical calculations, string manipulations, web programming, and many more.

Packages

Python package is a collection of related modules. You can either import a package or create your own.

List

You can think of a list as a collection of values. Declared in the CSV (Comma-Separated Values) format and delimit using square brackets:

```
life = ['love', 'wisdom', 'anxiety'];
arity = [1,2,3];
```

Notice that you do not declare the type for the list either. A list may also contain elements of different types, and the indexing begins at 0:

```
person = ['firstname', 21];
print(person[1])
```

```
Output: 21
```

Tuple

A tuple is like a list, but it is immutable (you cannot change its values). The following code will raise a TypeError.

```
pizza = ('base', 'sauce', 'cheese', 'mushroom');
pizza[3] = 'jalapeno'
```

Dictionary

A dictionary is a collection of key-value pairs. Declare it using curly braces, and commas to separate key-value pairs. Also, separate values from keys using a colon (:)

```
student = {'Name': 'Abc', 'Age': 21}
print(student['Age'])
```

```
Output: 21
```

Comments and Docstrings

Declare comments using an octothorpe (#). However, Python does not support multi-line comments. Also, docstrings are documentation strings that help explain the code.

```
# This is a comment
```

```
"""
This is a docstring
"""
```

Python has a lot of other constructs. These include control structures, functions, exceptions, etc.

More Python Learning Resources

You can go to available online education services to search for Python courses. For example, you can go to `coursera.com` and search 'Python', it will provides courses from beginners to advanced levels. Similar online education services are available such as edX, Udemy, Datacamp, Codecademy, Udacity, etc.

Features of Python

The features of Python is summarized in Figure CS-ML-00199.2.

Python is very easy to learn and understand; using this Python tutorial, any beginner can understand the basics of Python. It is interpreted(executed) line by line. This makes it easy to test and debug. The Python programming language supports classes and objects. The language and its source code are available to the public for free; there is no need to buy a costly license. Since it is open-source, you can run Python on Windows, Mac, Linux or any other platform. Your programs will work without needing to the changed for every machine. You can use it to develop a GUI (Graphical User Interface). One way to do this is through Tkinter. Python provides you with a large standard library. You can use it to implement a variety of functions without needing to reinvent the wheel every time. Just pick the code you need and continue. This lets you focus on other important tasks.

Tasks in Machine Learning Using Python

With Python Machine Learning, we divide the tasks of Machine Learning Algorithms in Python into two broad categories- Supervised and Unsupervised.

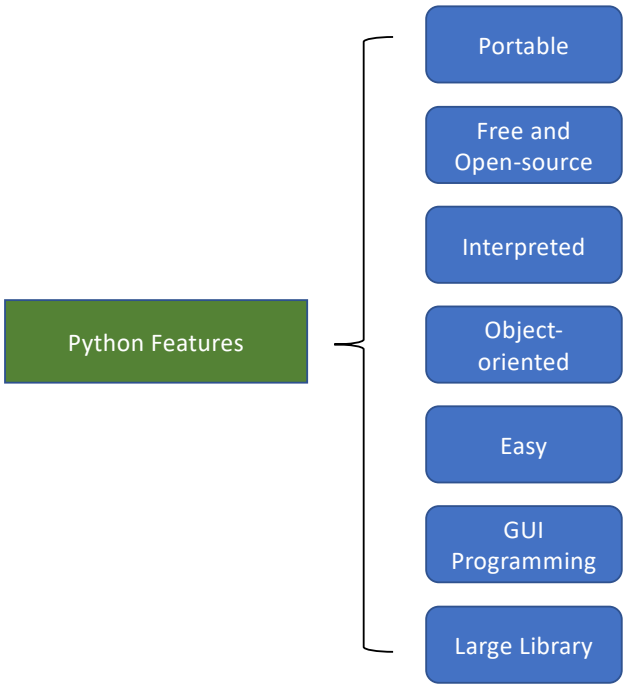


Figure CS-ML-00199.2
Summary of features of Python

Supervised Learning

Here, a learning signal/ feedback is available to the system; we give it to sample data to learn from. The computer holds example inputs and desired outputs with the goal of learning a general rule that maps inputs to outputs. One such example of Python Machine Learning will be to search for images on Facebook using keywords centered around the contents of the image. Under Supervised Learning, we have the following kinds of Python machine Learning-

- Semi-Supervised Learning- The computer receives an incomplete training signal. This is a training set with some target outputs missing.
- Active Learning- The computer can secure training labels for only some instances. It also needs to make an optimal choice of objects to secure labels.
- Reinforcement Learning- In this, the training data comes as feedback on how a program acts in a dynamic environment. Examples of this include driving a vehicle or playing against an opponent.

Steps involved in Supervised Machine Learning-

- training, and
- testing.

Among many Supervised Machine Learning Algorithms for beginners we observe, here we list some- Let's discuss Machine Learning Applications

- Decision trees
- Support Vector Machines
- Naïve Bayes
- k-nearest neighbor
- Linear regression

Unsupervised Learning

In unsupervised learning, the Python Machine Learning Algorithm receives no labels; we only give the machine a set of inputs. It must rely on itself to find structure in its input. This kind of learning can be a goal or a means toward future learning. We can classify unsupervised learning as-

- Clustering- The act of grouping data inherently. One example of this will be to group consumers by their shopping habits so they can target the right consumers to advertise.
- Association- In association, we identify rules explaining large sets of our data. One example will be to associate books around author/ category.

Of the many Unsupervised Machine Learning Algorithms, we observe, here are a couple-

- K-means clustering
- Hierarchical clustering

Steps in Python Machine Learning

We follow the following steps in Machine Learning Using Python-

1. Collecting data.
2. Filtering data.
3. Analyzing data.
4. Training algorithms.
5. Testing algorithms.
6. Using algorithms for future predictions.

Deliverable

Practice python programming environment (refer to lab CS-ML-00001) and understand the basic data science and python-based data processing concepts. For more details, please refer to the related information and resource.

Related Information and Resource

```
Python Tutorials:  https://data-flair.training/
```

“nobreak

Lab Assessment

Students should perform a self-assessment.