

LAHORE UNIVERSITY OF MANAGEMENT SCIENCES  
Syed Babar Ali School of Science and Engineering

EE514/CS535 Machine Learning  
Spring Semester 2021

Programming Assignment 1 – Introduction to Python

Issued: Tuesday 26 January, 2021

---

**Total Marks:** 100

**Submission:** 11:55 pm, Tuesday 02 February, 2021.

---

## Goal

The goal of this assignment is to set up python installation and perform some basic operations in Python.

## Instructions

- Submit your code both as notebook file (.ipynb) and python script (.py) on LMS. The name of both files should be your roll number. Failing to submit any one of them will result in the reduction of marks.
- All tasks should be implemented in the same file.
- The code MUST be implemented independently. Any plagiarism or cheating of work from others or the internet will be immediately referred to the DC.
- Late submissions: 10% penalty per day for 3 days after due date.

## Python Installation

### Setting up Anaconda

Anaconda is an open-source distribution of Python and R programming languages for scientific computing. Installing Anaconda will automatically install Python and the necessary packages for development and analysis. Click [here](#) to proceed to the official Anaconda page. Download the installer compatible with your system to install a fresh copy of Anaconda. Run the installer and follow the instructions there. Additionally, the following two resources present detailed instructions for installation:

- [Anaconda for Windows](#)
- [Anaconda for Mac](#)

## Verifying Installation

You may use any text editor to write Python code. One such text editor is available [here](#). However, be sure to save your code file with .py extension. For instance, your filename should read filename.py for it to be recognized as python file. To verify your installation, we will print the version of a few packages in the console. Write the following code in a file and save it:

```
import scipy
import numpy

print("scipy: %s" % scipy.__version__)
print("numpy: %s" % numpy.__version__)
```

Execute the code by running the following commands in the Terminal (Mac users) or Conda Prompt (Windows users). Make sure to first change the directory in which you are.

For Windows, change directory by typing in (replace "directory-path" with your own directory path where you saved the file):

```
cd "directory-path"
```

Run the file in that directory by typing in the command:

```
python filename.py
```

Note: the above command assumes you named your file as filename.py

## Checking Libraries

Before checking the libraries, we will also install one more library. Run the following command in your terminal:

```
pip install python-mnist
```

There should be a confirmation message telling you that the library is installed successfully.

Now, make a Python file with your text editor and run the following code to check if the relevant libraries are installed on your machine:

```
import numpy as np
import scipy as sp
from scipy.linalg import null_space
from numpy.linalg import inv
from matplotlib.image import imread
import matplotlib.pyplot as plt
import pandas as pd
import math
from scipy import stats
import random
from numpy.random import seed
from numpy.random import rand
from numpy.random import randn
```

```
from scipy.stats import pearsonr
from scipy.stats import ttest_ind
from scipy.stats import mannwhitneyu
from mnist import MNIST
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVC
from sklearn import metrics
from skimage.feature import hog
from sklearn.decomposition import PCA
from scipy.io import wavfile
from scipy.io.wavfile import write
import wave
import sys
```

Let the TA know if any if there is an error in running this code or if any of the libraries is not installed.

## Task 1: Lists, Dictionaries, Tuples (10 Marks)

### Lists

Given a list:

```
nums = [3, 4, 7, 8, 15]
```

Make another list named *cubes* and append the cubes of the given list in this list and print it.

### Dictionaries

You are given an empty dictionary:

```
dic = {}
```

- Add the following data to the dictionary: 'person': 2, 'cat': 4, 'spider': 8, 'horse': 4 as key value pairs.
- Use the 'items' method to loop over the dictionary and print the animals and their corresponding legs.
- Sum the legs of each animal, and print the total at the end.

### Tuples

Given the following tuple:

```
D = (1, 15, 4, [5, 10])
```

- Change the value in the list from '5' to '3'.
- Delete the tuple D.

Given another tuple:

```
E = ('a', 'p', 'p', 'l', 'e')
```

- Print the number of occurrences of 'p' in tuple E.
- Print the index of 'l' in tuple E.

## Task 2: Numpy (15 marks)

You should use all built in functions of numpy in this task, a list is available [here](#)

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

```
z = np.array([1, 0, 1])
```

- Convert matrix M into numpy array
- Use slicing to pull out the subarray consisting of the first 2 rows and columns 1 and 2. Store it in b which is a numpy array of shape (2, 2).
- Create an empty matrix 'y' with the same shape as 'M'.
- Add the vector z to each column of the matrix M with an explicit loop and store it in y.

Given:

```
A = np.array([[1,2],[3,4]])
```

```
B = np.array([[5,6],[7,8]])
```

```
v = np.array([9,10])
```

- Add the two matrices A and B.
- Multiply the two matrices A and B.
- Take the element wise square root of matrix A.
- Take the dot product of the matrix A and vector v.
- Compute sum of each column of A.
- Print the transpose of B.

## Task 3: Functions and For Loops (10 marks)

### Function

Declare a function *Compute* that takes two arguments: distance and time, and use it to calculate velocity.

## Forloop

Declare a list *even* that contains all even numbers up till 16. Declare a function *sum* that takes the list as an argument and calculates the sum of all entries using a for loop.

## Task 4: Matplotlib (15 marks)

Import the plotting function by the command:

```
import matplotlib.pyplot as plt
```

### Plotting a single line

Compute the  $x$  and  $y$  coordinates for points on a sine curve and plot the points using matplotlib. Use the function `plt.show()`

### Plotting multiple lines

Compute the  $x$  and  $y$  coordinates for points on sine and cosine curves and plot them on the same graph using matplotlib. Add  $x$  and  $y$  labels to the graph as well.

### Subplots

Compute the  $x$  and  $y$  coordinates for points on sine and cosine curves. Set up a subplot grid that has height 2 and width 1, and set the first such subplot as active. Plot the sine and cosine graphs.

*Hint: Use the `plt.subplot()` function*

## Task 5: Pandas DataFrame (15 marks)

### Making a DataFrame

Create a dataframe *pd* that contains 5 rows and 4 columns, similar to the one given below:

Col1	Col2	Col3	Col4
1	6	7	7
2	7	78	5
3	5	78	707
4	5	18	60
5	8	88	4

- Print only the first two rows of the dataframe.
- Print the second column.
- Change the name of the third column from “Col3” to “XYZ”.
- Add a new column to the dataframe and name it “Sum”.
- Sum the entries of each row and add the result in the column “Sum”.

## Task 6: Image Manipulation (20 marks)

### Importing Libraries

Run the following commands to import libraries

```
import matplotlib
from mpl_toolkits import mplot3d
from matplotlib import pyplot as plt
from matplotlib import cm
from matplotlib import image as mpimg
from matplotlib.pyplot import figure
%matplotlib inline
import seaborn as sns
import numpy as np
import matplotlib.pyplot as pl
from matplotlib.colors import ListedColormap
```

### Loading an Image

Read image from file to memory using the given command:

```
img = np.array(mpimg.imread('path'))
```

Replace 'path' in the above command with the path of your image e.g. *intro/cat.jpg*

Now run the command:

```
img.setflags(write=1)
```

This allows us to manipulate the image

### Tasks

- Display your image using the [plt.imshow](#) function.
- Crop the image.
- Create 50 randomly placed markers on your image. You can check the following [link](#) for learning about markers.
- Carry out color analysis by accessing the RGB values and plotting them. Use the *seaborn* library.
- View only the red values.  
*Hint: Set the blue and green values to 0*

## Task 7: Video Manipulation (15 marks)

### Importing Libraries

Run the following commands to import libraries

```
import cv2
import numpy as np
```

## Loading a Video

Create a VideoCapture object and read the input from camera:

```
video = cv2.VideoCapture(0)
```

## Tasks

- Check if the camera is open.
- Read the video from camera feed.
- Write the video file into memory using the [videoWriter](#) function.