BU.330.775 Machine Learning: Design and Deployment

**Lab 1. Python Tutorial**

Learning Goal: practice python basics and NumPy package

We will use Google CoLab, a popular online programming environment for Python. Please refer to the introduction page for details: <https://colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity_intro_to_tensorflow_for_deep_learning/l01c01_introduction_to_colab_and_python.ipynb>

If you prefer to install a Python programming environment on your own machine, please refer to lab 1 supplement document.

1. Open Google CoLab in your browser (Chrome may have issue with Copy/Paste function), and create a new notebook

Graphical user interface, text, application, email, website

Description automatically generated

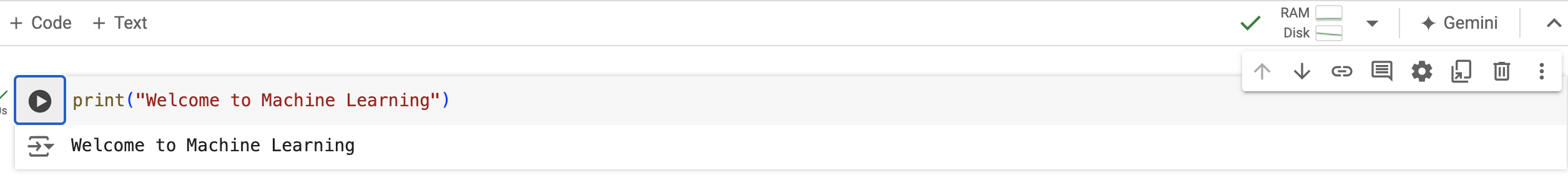
Now you will see an empty Python notebook in your browser.

1. As the first step of learning any programming language, let’s first try the print function of Python. In a cell, type in

print("Welcome to Machine Learning")

and then click the icon Icon

Description automatically generated with medium confidence to execute the Python code in the cell.



1. Python supports two types of numbers - integers and floating-point numbers. In the next cell(s), type in the following and execute.

myint = 330775

print(myint)

myfloat = 3.1415

print(myfloat)

1. Strings are defined either with single quotes or double quotes. In the next cell, type in the following and execute.

myFirstString = '330775'

print(myFirstString)

mySecondString = "machine learning"

print(mySecondString)

1. Then let’s do some operations on these numbers and strings. In the next cell(s), type in the following and execute.

addition = myint + myfloat

print(addition)

subtraction = myint - myfloat

print(subtraction)

multiplication = myint \* myfloat

print(multiplication)

division = myint / myfloat

print(division)

mod = myint % myfloat

print(mod)

concatenate1 = myFirstString + mySecondString

print(concatenate1)

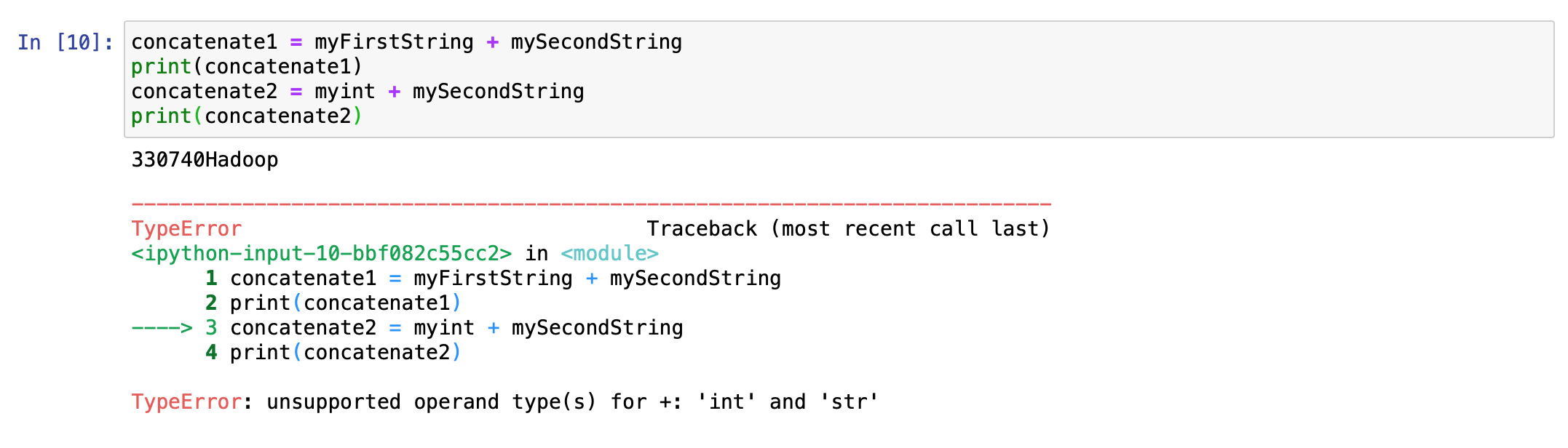
myFirstString += mySecondString

print(myFirstString)

concatenate2 = myint + mySecondString

print(concatenate2)

*What do you find out?*



1. Python supports arrays as lists. Next let’s define a list of numbers, and a list of strings. In the next cell(s), type in the following and execute.

intList = [330, 775]

stringList = []

stringList.append("machine")

stringList.append("learning")

print(intList[0])

print(intList[1])

print(stringList)

1. Python uses boolean variables to evaluate conditions. The boolean values True and False are returned when an expression is compared or evaluated. In the next cell, type in the following and execute.

name = "John"

age = 23

print(name == "John")

print(age > 25)

Python has if conditional statement, and use elif for elseif. Type the following example in the next cell and execute.

score = 85

if score >= 90:

print("A")

elif score >=80:

print("B")

else:

print("C")

1. Let’s try some useful for loops in Python. In the next cell(s), type in the following and execute.

for x in range(5):

print(x)

for x in range(1, 5):

print(x)

for x in stringList:

print(x)

1. (Optional) For advanced students, you can also use enumerate() for loops in Python with a result as a combination of index and value together. Here is a comparison, without enumerate():

seq = ['one', 'two', 'three']

for i in range(len(seq)):

print(i)

print(seq[i])

with enumerate():

seq = ['one', 'two', 'three']

for i, element in enumerate(seq):

print(i)

print(element)

For more examples: <https://book.pythontips.com/en/latest/enumerate.html>

1. Functions in python are defined using the block keyword "def", followed by the function's name and arguments if any. Define a function for me to covert total score to letter grade as following:

def convert2letter(totalScore):

if totalScore >= 90:

return "A"

elif totalScore >= 80:

return "B"

else:

return "C"

and then test it using

johnScore = 80

print(convert2letter(johnScore))

1. An f-string (formatted string literal) in Python is a way to embed expressions inside string literals, using curly braces {}. **Introduced in Python 3.6**, f-strings are prefixed with the letter f or F, making it easier and more readable to include variables or expressions inside strings (provided by ChatGPT-4o).

A screenshot of a computer

Description automatically generated

Let’s try a few f-string examples using Google Colab’s “generate with AI” function. Note that you need a Gmail account for this function.

1. A dictionary is a data type similar to arrays, but works with keys and values instead of indexes. For example, I can maintain a students’ scorebook in the following example.

scoreBook={}

scoreBook["John"]=80

scoreBook["Jack"]=75

scoreBook["Jill"]=92

print(scorebook)

# F-string with dictionary

print(f"Jack receives a {convert2letter(scoreBook['Jack'])} from machine learning.")

1. We will introduce the first Python package “NumPy”, a popular Python library used for numerical computing, particularly for working with arrays and matrices. It provides support for multi-dimensional arrays and a variety of mathematical functions to operate on these arrays efficiently. NumPy is widely used in scientific computing, data analysis, and machine learning because of its ability to handle large datasets and perform complex mathematical operations quickly (provided by ChatGPT-4o).

Let’s first import the package.

import numpy as np

1. In NumPy, each dimension is called an **axis**. The number of axes is called the **rank**. For example, a 3x4 matrix is an array of rank 2, or it is 2-dimensional. The first axis has length 3, the second has length 4. An array's list of axis lengths is called the **shape** of the array. For example, the above matrix's shape is (3, 4). The size of an array is the total number of elements, which is the product of all axis lengths (e.g. 3\*4=12). Let’s see the example.

a = np.zeros((3,4))

print(a)

print(a.ndim)

print(a.shape)

print(a.size)

1. The linspace function in NumPy returns an array containing a specific number of points evenly distributed between two values.

np.linspace(1, 5, 9)

1. A number of functions are available in NumPy's random module to create arrays initialized with random values. For example, here are a 3x4 matrix initialized with random floats between 0 and 1 (uniform distribution), and a 3x4 matrix containing random floats sampled from a univariate normal distribution (Gaussian distribution) of mean 0 and variance 1:

np.random.rand(3,4)

np.random.randn(3,4)

**Homework Question 1 (6pt)**: Explore, by search or consulting ChatGPT (please specify your source in your submission), the following statistical functions of NumPy arrays:

* 1. mean()
  2. max()
  3. min()
  4. sum()
  5. std()
  6. var()

In a “Text” cell, describe each of the functions.

In a “Code” cell, provide the correct Python code for these functions on this array:

homework = np.array([[1,2,3,4], [10, 20, 30, 40], [60, 70, 80, 90]])

1. Let’s play with the fancy indexing of NumPy arrays:

a = np.array([1, 2, 3, 4, 10, 20, 30])

print(a[3])

print(a[1:3])

print(a[1:5:2])

print(a[2:-1])

print(a[:2])

Tricky? Here is another one for you to explore as homework.

**Homework Question 2 (1pt)**: Explore, by simply executing and interpreting or consulting ChatGPT, what the following indexing does. Explain it in a “Text” cell.

homework[(0,2), 1:3]

1. Finally, let’s rename the notebook. Click the name of the notebook (as shown below), and change it to a name you choose.

A screenshot of a computer

Description automatically generated

You can download the .ipynb file to your local directory by clicking “File->Download”

A screenshot of a computer

Description automatically generated

**Submission**: Complete all the lab steps and the 2 homework questions. Save your file as homework1\_yourname.ipynb and submit on Canvas by the beginning of class 2.