

Machine Learning Foundations: A Comprehensive Guide from Python to Mathematics

Mr. Saeed Ullah

2025-08-01

Table of contents

Preface	3
Attention to Details	3
Do Not Copy-Paste	3
A Note for Smarties	3
1 1 Windows OS Setup	5
2 Introduction to Python	6
2.1 Python Basics:	7
2.1.1 Comments in python	8
2.1.2 Numbers & Math in python	8
2.1.3 Variable & Names	9
2.1.4 Escape Sequences	11
2.1.5 Python modules	12
References	15

Preface

This book is for those who are fully beginner and want to learn how can they become master in machine learning from no-coding experience to become proficient ML Engineer of the future. This is not just a book but a journey of a machine learning engineer and what he learn and explore throughout his journey from going python to advanced topic like deep learning algorithm and the mathematics behind it. So be patience and read it step-by-step and you will be rewarded with the knowledge that will make you from ordinary to extra-ordinary person Inshallah!.

Attention to Details

Key that differ a bad program from a good one is to attention to the detail of the problem. In fact, its applies to every proficient. Those who aren't paying attention may end up with a lot of buggy code and will not be able to solve it proficiently and will always do dwelling on internet to solve only a simple problem.

Do Not Copy-Paste

Do not copy-paste any of the coding example. Practice it by yourself and play around with it a little bit, Tweak it so that you understand how it works.

A Note for Smarties

If any of the reader think that i am insulting him then don't read this book because its for those who is beginner and want to start their career in machine learning or Artificial Inteligence. If you know everything then perhaps you should learn Lisp, which stands for (List Processing) programming language. I hear people who know everything really like Lisp.

Note

Engaging in real-world problem-solving and delving into research requires a significant time commitment.

Let's start from here...

1 1 Windows OS Setup

Install python from the following link or watch any youtube video on it to install it as it requires only vscode (editor) and python language installation: Certainly, Asad_Pro_3.1. Here's a paraphrased version of the instructions:

1. Visit [Notepad++](#), download the Notepad++ text editor, and install it without requiring administrator privileges.
2. Ensure easy access to Notepad++ by placing shortcuts on your desktop and/or in Quick Launch during the installation.
3. Launch PowerShell from the Start menu by searching for it and pressing Enter.
4. Create shortcuts for PowerShell on your desktop and/or Quick Launch for convenience.
5. Open your Terminal program, and if Python is not recognized, install it from Python.
6. Consider using ActiveState Python if administrative rights are an issue. If Python isn't recognized after installation, set the environment variable in PowerShell and restart it.
 - a. If you run python and it's not there (python is not recognized.), install it from <http://python.org/download>.
 - b. Make sure you install Python 2, not Python 3.
 - c. You may be better off with ActiveState Python, especially if you do not have administrative rights.
 - d. If after you install it python still isn't recognized, then in PowerShell enter this:
`[Environment]::SetEnvironmentVariable("Path", "$env:Path;C:", "User")`
 - e. Close PowerShell and then start it again to make sure Python now runs. If it doesn't, restart may be required.
7. Type `quit()` in Python to exit.
8. Return to the previous prompt in the Terminal and troubleshoot if necessary.
9. Learn how to create a directory in the Terminal.
10. Familiarize yourself with changing directories in the Terminal.
11. Utilize your text editor to create a file in the directory, then save it.
12. Switch back to the Terminal using keyboard shortcuts.
13. In the Terminal, check if you can list the directory to confirm the creation of your new file.

2 Introduction to Python

Scroll Progress Bar

Hello Readers, here's a quick, simple and *kid-friendly* introduction to Python:

Python is like a magic language that computers understand. It's not as tricky as some other languages, making it perfect for beginners, including kids!

What Python Can Do:

1. **Talking to Computers:** Imagine Python as a way to talk to computers in a language they understand. You can tell them what to do, and they'll do it!
2. **Making Games:** Have you ever wanted to create your own video game? Python can help you do that. You can tell the computer how the game should work, and soon you'll have your very own creation.
3. **Drawing and Art:** Python isn't just for words; it can draw too! You can create beautiful pictures and designs with Python. It's like having a digital paintbrush.
4. **Solving Puzzles:** Python is like a superhero when it comes to solving problems. You can use it to figure out puzzles and challenges. It's your own coding superhero!
5. **Talking to Robots:** Robots are cool, right? With Python, you can tell robots what to do. It's like being a commander for your own robot army!

Real-World Examples:

1. **Weather Helper:** Python can help you check the weather. You can create a program that tells you if it's going to be sunny or rainy. No more surprises when you step outside!
2. **Homework Assistant:** Python can be your homework helper. You can make a program that solves math problems or checks your spelling. It's like having a smart friend by your side.



Caution

Don't take its literal meaning to solve your homework problem. Instead, take it as your mentor to ask for explanation of your problem and ask it to give the working of the underlying solutions.

3. **Virtual Pet:** Ever wanted a pet dragon or a virtual friend? Python can help you create a virtual pet that you can feed, play with, and take care of. It's like having a magical pet on your computer!
4. **Treasure Hunt Game:** Python can turn your computer into a treasure map. Create a game where you search for hidden treasures by giving commands to your Python friend. X marks the spot!

Remember, Python is like your secret code to make cool things happen on your computer. Have fun coding!

! Important

Finally the purpose of the above setup was to do four things reliably while you work on the exercises:

1. Write exercise in your favourite text editor (mine VScode).
2. Run the exercise.
3. Fix the error when arise.
4. Repeat

Anything else will only confuse you so stick to the plan.

2.1 Python Basics:

Open search bar and open one of the following:

- Command Prompt (cmd) for short.
- PowerShell
- IDLE Python shell
- Open your favourite Editor and write the following code into it and then save it as hello.py (python programs can be run only if its in **.py** extension)

Run your python script in directory where you have store your .py python file just open cmd there and run the command: `python filename.py`

```
print("Yay! i am writing my first program") ①
print('i am printing') ②
(' ' , "") ③
```

- ① `print` is called a function which will be explain in the letter chapters. so don't stuck on these programming lingo for now.

- ② As you can see between those yellow brackets, text can be enclosed in both single and double quote.
- ③ These quotes tells the python that we are writing string and strings are like plain text to python, its that simple.

Tip

Also run your code in python Visual Compiler which explain and run your code step by step to show you what happens under the hood. Here's the link: [Python Visual Compiler](#)

2.1.1 Comments in python

Comments are used to tell what the specific part of your program do and its like an instruction box for your line of code. It can also be used to disable your code. so this it won't run.

```
print('printing nothing') #single line comment ①
'''Multiline comments''' ②
```

- ① This is single line of comment. Here with pound character i can write anything just to explain this piece of code not to execute it
- ② You can also use multiline comments with the help of 3 single quotes which is used to write multiple line without specifying any # hash or pound character everytime you start new line of text.

2.1.2 Numbers & Math in python

Table 2.1: Most Commonly use Mathematics symbols

Symbols	Names
+	Plus
-	Minus
/	Slash
*	Asterisk
%	Percent
<	Less-than
>	Greater-than
<=	Less-than-equal
>=	Greater-than-equal

2.1.3 Variable & Names

First let's understand **Variable** in simple terms:

Imagine you have a box, and you give it a name, like `number_box`. Now, you can put a number in it, let's say 5. So, `number_box` holds the value 5.

```
number_box = 5
```

Here `number_box` is the variable and that number 5 is the value in it. You can also change it like below

```
number_box = 10 ①  
print(number_box) # Output: 10 ②
```

Now it contains 10 instead of 5. Variables are useful and they let you store and keep track of your numbers, text, and other things you will see in the advanced sections.

Quiz time:

Question 1: Write a Python code snippet to print the phrase "Hello, Python!"

Question 2: How can you print the result of multiplying two numbers, let's say 4 and 7, in Python?

Question 3: Explain what the end parameter in the print function is used for.

Question 4: Write a Python code snippet to print your name on one line and your age on the next line.

Question 5: If you want to print the following text on separate lines:

```
Programming  
is  
fun!
```

How would you achieve this using the print function?

Click to see answers...

1. `print("Hello, Python!")`
2. `print(4 * 7)`
3. The end parameter in the print function is used to specify the character that separates multiple printed items. By default, it is set to `\n`, which means a newline character. Newline is like when we reach the end of a line we press enter; that's exactly what it does, just to press enter at the end of a line.

```
4. print("Your Name")
5. print("Your Age")
6. print("Programming")
7. print("is")
8. print("fun!")
```

String formatting

We can also format our string output in different ways, First i will write code and then from the following you can see its explanation below:

```
my_name = 'Asad Pro Beta' # My digital persona
address = 'Peshawar, Pakistan'
my_age = 35 # Just kidding
my_height = 74 # inches

print(my_name+' ',address) ①
print(my_name+3*'.') ②
print('Your name is: {} my age: {}'.format(my_name,my_age)) ③
print('Your age is: {1} my name: {0}'.format(my_name, my_age)) ④
print(f'Your name is: {my_name} my age: {my_age}') ⑤
print(r'Your name is: {my_name} my age: {my_age}')
```

- ① This + sign merges (concatenate) two strings together and that white space is separate them
- ② First multiply dot with 3 which become (...) & then concatenate (merge) it with that string (my_name)
- ③ The format function in Python is like a template for creating sentences where you want to replace certain parts with specific values. Let me explain with an analogy: Imagine you have a fill-in-the-blank storybook. The storybook has sentences with blanks, and you have stickers with words on them. Each sticker has a different word that fits into one of the blanks.
- ④ We can also use index number, Index are like address of every variable where it stored on computer memory.
- ⑤ Putting f in the beginning act as a format function with which it identify variables.
- ⑥ Putting r in the beginning means that this string is raw which means nothing in other words, print everything as it is.

```
A,B,C,D,E = [i for i in range(ord('A'),ord('E')+1)] ①
a,b,c,d,e = [i for i in range(ord('a'),ord('e')+1)]
```

```

print(A,B,C,D,E,                                     ②
      a,b,c,d,e)

A,B,C,D,E = [65, 66, 67, 68, 69]                    ③

days = "Mon Tue Wed Thu Fri Sat Sun"
months = "Jan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug"    ④

print(days)
print(months)

```

- ① This is another way to assign list of values to multiple variable at once. (Don't bother we will cover **Lists** in later chapters)
- ② Remove that comma after 'E' and see what happens
- ③ Its the same as the first one above, I have just created the list dynamically in the first place.
- ④ Here you will see that \n character will act as an enter and printing will start from the new line. Copy and paste it in your editor and see its result on your own.

Click to explore print function thoroughly

More on [print](#)

2.1.4 Escape Sequences

Imagine you're writing a story on the computer, and you want to do some special things with your words. Escape sequences in Python are like magic codes you can use to make your words look different or do special tricks!

Look at the following table and try them out on your own:

Escape	What it does
\\	Backslash (\)
\'	Single-quote (')
\"	Double-quote (")
\a	ASCII bell (BEL)
\b	ASCII backspace (BS)
\f	ASCII formfeed (FF)
\n	ASCII linefeed (LF)
\N{name}	Character named name in the Unicode database
\r	ASCII carriage return (CR)

Escape	What it does
<code>\t</code>	ASCII horizontal tab (TAB)
<code>\uxxxx</code>	Character with 16-bit hex value <code>xxxx</code> (Unicode only)
<code>\Uxxxxxxxx</code>	Character with 32-bit hex value <code>xxxxxxxx</code> (Unicode only)
<code>\v</code>	ASCII vertical tab (VT)
<code>\ooo</code>	Character with octal value <code>oo</code>
<code>\xhh</code>	Character with hex value <code>hh</code>

Prompting People or Asking for Input

```
print('How old are you?')
age = input()
print(f'Your age : {age}')
```

`input()` is a method which shows you an empty placeholder for you to insert any value & when you insert then press enter so that value can be store in **age** variable.

`Input()` function is like an empty page; someone gives you and you write anything what comes to your mind.

2. Here, we have used the same [string formatting](#) as we use earlier.

2.1.5 Python modules

Let's see the code first...

```
import os
current_path = os.getcwd()
print(current_path)
```

- ① `os` is a built-in module of python which is used to do local operating system level operation like directory change, path of the folder and navigating through directories etc.
- ② This code just copy your path where you are running your file. Don't confuse yourself with that `os.getcwd()` for now you will soon understand this.

Module in simple terms

Think of these module **OS** this way. If python is an electrician then modules are its tool sets with every tool has their specific purpose like (pliers, voltage tester, wire stripper, multimeter etc) or you can say these modules are the features of python.

Asking Prompt and Passing Argument

```
from sys import argv
script_name, username = argv
print(f'Script name: {script_name}')
print(f'Username: {username}')
```

```
'''
Input:  python delete.py asad
Output: Script name: delete.py
        Username: asad
'''
```

1. This is the other way to import module(feature) which says import from the module **sys** the **argv** method (which is a small part of that module)
2. First open the terminal and navigate to your file where it is stored then run **python delete.py asad** here **delete.py** is my python file name write what's yours and **asad** is the argument like an extra input to catch it inside our program.

3

References