



Hands-on Machine Learning with Python

Implement Neural Network Solutions
with Scikit-learn and PyTorch

Ashwin Pajankar
Aditya Joshi

Apress®

Hands-on Machine Learning with Python

**Implement Neural Network Solutions
with Scikit-learn and PyTorch**

**Ashwin Pajankar
Aditya Joshi**

Apress®

Hands-on Machine Learning with Python: Implement Neural Network Solutions with Scikit-learn and PyTorch

Ashwin Pajankar
Nashik, Maharashtra, India

Aditya Joshi
Haldwani, Uttarakhand, India

ISBN-13 (pbk): 978-1-4842-7920-5
<https://doi.org/10.1007/978-1-4842-7921-2>

ISBN-13 (electronic): 978-1-4842-7921-2

Copyright © 2022 by Ashwin Pajankar and Aditya Joshi

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: James Markham
Coordinating Editor: Aditee Mirashi

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-7920-5. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

This book is dedicated to the memory of our teacher, Prof. Govindarajulu Regeti (July 9, 1945–March 18, 2021)



Popularly known to everyone as RGR, Prof. Govindarajulu obtained his B.Tech. in Electrical and Electronics Engineering from JNTU Kakinada. He also earned his M.Tech. and Ph.D. from IIT Kanpur. Prof. Govindarajulu was an early faculty member of IIIT Hyderabad and played a significant role in making IIIT Hyderabad a top-class institution that it grew to become today. He was by far the most loved and cheered for faculty member of the institute. He was full of energy to teach and full of old-fashioned charm. There is no doubt he cared for every student as an individual, taking care to know about and to guide them. He has taught, guided, and mentored many batches of students at IIIT Hyderabad (including one of the authors of the book, Ashwin Pajankar).

Table of Contents

About the Authors	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
Section 1: Python for Machine Learning	1
Chapter 1: Getting Started with Python 3 and Jupyter Notebook	3
Python 3 Programming Language.....	4
History of Python Programming Language.....	4
Where Python Is Used.....	5
Installing Python	6
Python on Linux Distributions.....	8
Python on macOS	8
Python Modes	8
Interactive Mode.....	9
Script Mode	15
Pip3 Utility.....	17
Scientific Python Ecosystem.....	18
Python Implementations and Distributions	19
Anaconda	20
Summary.....	21
Chapter 2: Getting Started with NumPy	23
Getting Started with NumPy.....	24
Multidimensional Ndarrays.....	26
Indexing of Ndarrays	26

TABLE OF CONTENTS

- Ndarray Properties 28
- NumPy Constants..... 29
- Summary..... 30
- Chapter 3: Introduction to Data Visualization..... 31**
- NumPy Routines for Ndarray Creation 31
- Matplotlib Data Visualization..... 34
- Summary..... 43
- Chapter 4: Introduction to Pandas..... 45**
- Pandas Basics..... 45
- Series in Pandas 46
- Properties of Series 47
- Pandas Dataframes..... 48
- Visualizing the Data in Dataframes 50
- Summary..... 61
- Section 2: Machine Learning Approaches 63**
- Chapter 5: Introduction to Machine Learning with Scikit-learn 65**
- Learning from Data 66
- Supervised Learning..... 66
- Unsupervised Learning..... 67
- Structure of a Machine Learning System..... 68
- Problem Understanding 69
- Data Collection 69
- Data Annotation and Data Preparation 70
- Data Wrangling 70
- Model Development, Training, and Evaluation 70
- Model Deployment..... 71
- Scikit-Learn 72
- Installing Scikit-Learn..... 72

Understanding the API.....	73
Your First Scikit-learn Experiment	75
Summary.....	77
Chapter 6: Preparing Data for Machine Learning	79
Types of Data Variables.....	79
Nominal Data.....	80
Ordinal Data.....	80
Interval Data	81
Ratio Data	81
Transformation	81
Transforming Nominal Attributes.....	82
Transforming Ordinal Attributes.....	84
Normalization.....	86
Min-Max Scaling.....	87
Standard Scaling	88
Preprocessing Text.....	89
Preparing NLTK.....	89
Five-Step NLP Pipeline	90
Preprocessing Images	94
Summary.....	97
Chapter 7: Supervised Learning Methods: Part 1	99
Linear Regression	99
Finding the Regression Line	100
Logistic Regression.....	107
Line vs. Curve for Expression Probability	108
Learning the Parameters	109
Logistic Regression Using Python	110
Visualizing the Decision Boundary	112

TABLE OF CONTENTS

- Decision Trees 114
 - Building a Decision Tree 114
 - Decision Tree in Python 116
- Summary..... 120
- Chapter 8: Tuning Supervised Learners..... 121**
- Training and Testing Processes..... 121
- Measures of Performance..... 122
 - Confusion Matrix 122
 - Precision..... 124
 - Accuracy..... 124
 - F-Measure 124
 - Performance Metrics in Python 125
- Cross Validation 128
 - Why Cross Validation? 129
 - Cross Validation in Python 129
- ROC Curve..... 131
- Overfitting and Regularization 134
 - Bias and Variance 138
 - Regularization..... 139
- Hyperparameter Tuning..... 142
 - Effect of Hyperparameters 143
- Summary..... 148
- Chapter 9: Supervised Learning Methods: Part 2..... 149**
- Naive Bayes 149
 - Bayes Theorem..... 150
 - Conditional Probability..... 150
 - How Naive Bayes Works 151
 - Multinomial Naive Bayes 151
 - Naive Bayes in Python 152

Support Vector Machines	155
How SVM Works	155
Nonlinear Classification.....	157
Kernel Trick in SVM.....	158
Support Vector Machines in Python.....	159
Summary.....	165
Chapter 10: Ensemble Learning Methods	167
Bagging and Random Forest.....	168
Random Forest in Python	170
Boosting	172
Boosting in Python.....	175
Stacking Ensemble	180
Stacking in Python.....	181
Summary.....	184
Chapter 11: Unsupervised Learning Methods.....	185
Dimensionality Reduction	185
Understanding the Curse of Dimensionality	186
Principal Component Analysis	187
Principal Component Analysis in Python	189
Clustering.....	192
Clustering Using K-Means	193
K-Means in Python	194
Frequent Pattern Mining	205
Market Basket Analysis	206
Frequent Pattern Mining in Python.....	207
Summary.....	211

Section 3: Neural Networks and Deep Learning 213

Chapter 12: Neural Network and PyTorch Basics 215

 Installing PyTorch..... 216

 PyTorch Basics 217

 Creating a Tensor..... 217

 Tensor Operations..... 219

 Perceptron 221

 Perceptron in Python 223

 Artificial Neural Networks 225

 Summary..... 226

Chapter 13: Feedforward Neural Networks 227

 Feedforward Neural Network..... 228

 Training Neural Networks 229

 Loss Functions..... 233

 ANN for Regression 234

 Activation Functions..... 239

 ReLU Activation Function..... 240

 Sigmoid Activation Function 240

 Tanh Activation Function 241

 Multilayer ANN 242

 NN Class in PyTorch..... 248

 Overfitting and Dropouts 251

 Classifying Handwritten Digits 253

 Summary..... 260

Chapter 14: Convolutional Neural Networks..... 261

 Convolution Operation..... 261

 Structure of a CNN 266

 Padding and Stride 267

 CNN in PyTorch..... 268

Image Classification Using CNN.....	271
What Did the Model Learn?	282
Deep Networks of CNN	284
Summary.....	284
Chapter 15: Recurrent Neural Networks.....	285
Recurrent Unit.....	286
Types of RNN.....	288
One to One	288
One to Many	289
Many to One	289
Many to Many.....	289
RNN in Python	290
Long Short-Term Memory	291
LSTM Cell.....	292
Time Series Prediction	293
Gated Recurrent Unit.....	304
Summary.....	305
Chapter 16: Bringing It All Together	307
Data Science Life Cycle	308
CRISP-DM Process	309
How ML Applications Are Served	312
Learning with an Example	312
Defining the Problem.....	313
Data	313
Preparing the Model	319
Serializing for Future Predictions	325
Hosting the Model	327
What's Next.....	328
Index.....	329

About the Authors

Ashwin Pajankar is an author, an online instructor, a content creator, and a YouTuber. He has earned a Bachelor of Engineering from SGGSI&T Nanded and an M.Tech. in Computer Science and Engineering from IIIT Hyderabad. He was introduced to the amazing world of electronics and computer programming at the age of seven. BASIC is the very first programming language he learned. He has a lot of experience in programming with Assembly Language, C, C++, Visual Basic, Java, Shell Scripting, Python, SQL, and JavaScript. He also loves to work with single-board computers and microcontrollers like Raspberry Pi, Banana Pro, Arduino, BBC Microbit, and ESP32.

He is currently focusing on developing his YouTube channel on computer programming, electronics, and microcontrollers.

Aditya Joshi is a machine learning engineer who's worked in data science and ML teams of early to mid-stage startups. He has earned a Bachelor of Engineering from Pune University and an M.S. in Computer Science and Engineering from IIIT Hyderabad. He became interested in machine learning during his masters and got associated with the Search and Information Extraction Lab at IIIT Hyderabad. He loves to teach, and he has been involved in training workshops, meetups, and short courses.

About the Technical Reviewer

Joos Korstanje is a data scientist, with over five years of industry experience in developing machine learning tools, of which a large part is forecasting models. He currently works at Disneyland Paris where he develops machine learning for a variety of tools.

Acknowledgments

I would like to express my gratitude toward Aditya Joshi, my junior from IIIT Hyderabad and now an esteemed colleague who has written the major and the most important section of this book. I also wish to thank my mentors from Apress, Celestin, Aditee, James Markham, and the editorial team. I wish to thank the reviewers who helped me make this book better. I also thank Prof. Govindrajulu's family – Srinivas (son) and Amy (daughter-in-law) – for allowing me to dedicate this book to his memory and sharing his biographical information and his photograph for publication.

—Ashwin Pajankar

My work on this book started with a lot of encouragement and support from my father, Ashok Kumar Joshi, who couldn't live long enough to see it till completion. I am extremely grateful to friends and family – especially my mother, Bhavana Joshi, and many others, whose constant support was the catalyst to help me work on this project. I also want to extend my heartiest thanks to my wife, Neha Pandey, who was supportive and patient enough when I extended my work especially during weekends. I would like to thank Ashwin Pajankar, who's been not just a coauthor but a guide throughout this journey. I'd also like to extend my gratitude to the Innomatics team, Kalpana Katiki Reddy, Vishwanath Nyathani, and Raghuram Aduri, for giving me opportunities to interact with hundreds of students who are learning data science and machine learning. I'd also like to thank Akshaj Verma for his support with code examples in one of the advanced chapters. I also thank the editorial team at Apress, especially Celestin Suresh John, Aditee Mirashi, James Markham, and everyone who was involved in the process.

—Aditya Joshi

Introduction

We have long been planning to collaborate and write a book on machine learning. This field has grown and expanded immensely since we started learning these topics almost a decade ago. We realized that, as lifelong learners ourselves, the initial few steps in any field require a much clearer source that shows a path clearly. This also requires a crisp set of explanation and occasional ideas to expand the learning experience by reading, learning, and utilizing what you have learned. We have used Python for a long duration in our academic life and professional careers in software development, data science, and machine learning. Through this book, we have made a very humble attempt to write a step-by-step guide on the topic of machine learning for absolute beginners. Every chapter of the book has the explanation of the concepts used, code examples, explanation of the code examples, and screenshots of the outputs.

The first chapter covers the setup of the Python environment on different platforms. The second chapter covers NumPy and Numpy arrays. The third chapter explores visualization with Matplotlib. The fourth chapter introduces us to the Pandas data science library. All these initial chapters build the programming and basic data crunching foundations that are one of the prerequisites for learning machine learning.

The next section discusses traditional machine learning approaches. In Chapter 5, we start with a bird's-eye view of the field of machine learning followed by the installation of Scikit-learn and a short and quick example of a machine learning solution with Scikit-learn. Chapter 6 elaborates methods to help you understand and transform structural, textual, and image data into the format that's acceptable by machine learning libraries. In Chapter 7, we introduce supervised learning methods, starting with linear regression for regression problems and logistic regression and decision trees for classification problems. In each of the experiments, we also show how to plot visualizations that the algorithm has learned with the use of decision boundary plots. The eighth chapter ponders over further fine-tuning of machine learning models. We explain some ideas for measuring the performance of the models, issues of overfitting and underfitting, and approaches for handling such issues and improving the model performance. The ninth chapter continues the discussion of supervised learning methods especially focusing on naive Bayes and Support Vector Machines. The tenth

INTRODUCTION

chapter explains ensemble learning methods, which are the solutions that combine multiple simpler models to produce a performance better than what they might offer individually. In the eleventh chapter, we discuss unsupervised learning methods, specifically focusing on dimensionality reduction, clustering, and frequent pattern mining methods. Each part contains a complete example of implementing the discussed methods using Scikit-learn.

The last section begins with introducing the basic ideas of neural network and deep learning in the twelfth chapter. We introduce a highly popular open source machine learning framework, PyTorch, that will be used in the examples in the subsequent chapters. The thirteenth chapter begins with the explanation of artificial neural networks and thoroughly discusses the theoretical foundations of feedforward and backpropagation, followed by a short discussion on loss functions and an example of a simple neural network. In the second half, we explain how to create a multilayer neural network that is capable of identifying handwritten digits. In the fourteenth chapter, we discuss convolutional neural networks and work through an example for image classification. The fifteenth chapter discusses recurrent neural networks and walks you through a sequence modeling problem. In the final, sixteenth chapter, we discuss strategies for planning, managing, and engineering machine learning and data science projects. We also discuss a short end-to-end example of sentiment analysis using deep learning.

If you are new to the subject, we highly encourage you to follow the chapters sequentially as the ideas build upon each other. Follow through all the code sections, and feel free to modify and tweak the code structure, datasets, and hyperparameters. If you already know some of the topics, feel free to skip to the topics of your interest and examine the relevant sections thoroughly. We wish you the best for your learning experience.

SECTION 1

Python for Machine Learning

CHAPTER 1

Getting Started with Python 3 and Jupyter Notebook

I hope that all of you have read the introduction and the table of contents. This is very important because if you are a complete beginner, please do not skip this chapter. The entire field of machine learning and artificial intelligence requires solid knowledge of the tools and frameworks used in the area. This chapter serves as the foundational chapter for Python programming for machine learning that we will cover in this book. It introduces the novice readers to the Python Programming Language, Scientific Python Ecosystem, and Jupyter Notebook for Python programming.

The following is the list of topics that we will learn in this chapter:

- Python 3 Programming Language
- Installing Python
- Python Modes
- Pip3 Utility
- Scientific Python Ecosystem
- Python Implementations and Distributions

After studying this chapter, we will be comfortable with the installation, running programs, and Jupyter notebook on Windows and Debian Linux.

Python 3 Programming Language

Python 3 is a modern programming language. It has features of object-oriented and procedural programming. It runs on a variety of platforms. The most readily available platforms for common readers are macOS, Windows, and various Linux distributions. Python runs on all of them. And the major advantage is that the code written on one platform runs on the other platform without any major changes to the code (except for the platform-specific code). You can read more about Python at www.python.org/.

History of Python Programming Language

The ABC language is the predecessor of Python Programming Language. The ABC language was inspired by ALGOL 68 and SETL programming languages. The Python Programming Language was created by Guido Van Rossum as a side project during Christmas vacations in the late 1980s. He was working at the National Research Institute for Mathematics and Computer Science (Centrum Wiskunde & Informatica). Van Rossum was a graduate of the University of Amsterdam in computer science. He has worked for Google and Dropbox. Currently, he works for Microsoft.

Python has two major and incompatible versions: Python 2 and Python 3. Python 2 is now not under active development and maintenance. The entire (programming) world is gradually switching to Python 3 from Python 2. For all the demonstrations in this book, we will use Python 3. Whenever we use the word **Python**, it will mean **Python 3** from now onward.

Philosophy of Python Programming Language

The philosophy of Python Programming Language is known as **The Zen of Python**, and it can be accessed at www.python.org/dev/peps/pep-0020/. The following are the points from that PEP (Python Enhancement Proposal). A few are funny.

1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.

6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.
10. Errors should never pass silently.
11. Unless explicitly silenced.
12. In the face of ambiguity, refuse the temptation to guess.
13. There should be one – and preferably only one – obvious way to do it.
14. Although that way may not be obvious at first unless you're Dutch.
15. Now is better than never.
16. Although never is often better than **right** now.
17. If the implementation is hard to explain, it's a bad idea.
18. If the implementation is easy to explain, it may be a good idea.
19. Namespaces are one honking great idea – let's do more of those!

These are general philosophical guidelines that influenced the development of the Python Programming Language over decades and continue to do so.

Where Python Is Used

Python is used in a variety of applications. A few are

- 1) Education
- 2) Automation
- 3) Scientific Computing
- 4) Computer Vision
- 5) Animation
- 6) IoT

- 7) Web Development
- 8) Desktop and Mobile Applications
- 9) Administration

We can read all the applications in detail at www.python.org/about/apps/. Many organizations have used Python to create applications. We can read all these success stories at www.python.org/success-stories/. Now, let's begin with our very own one.

Installing Python

We will learn how to install Python 3 on Windows in detail. Visit www.python.org and hover the mouse pointer over the **Downloads** options. It will open the downloads menu. It will show the appropriate option depending on your OS. In our case, it will show option for downloading on Windows. Download the file. It is an executable installation file. In my case, it downloads a 64-bit version of the installable file. If you are using other architecture (e.g., 32 bit), then it will download the appropriate file accordingly. Figure 1-1 shows the Python 3 download for Windows.

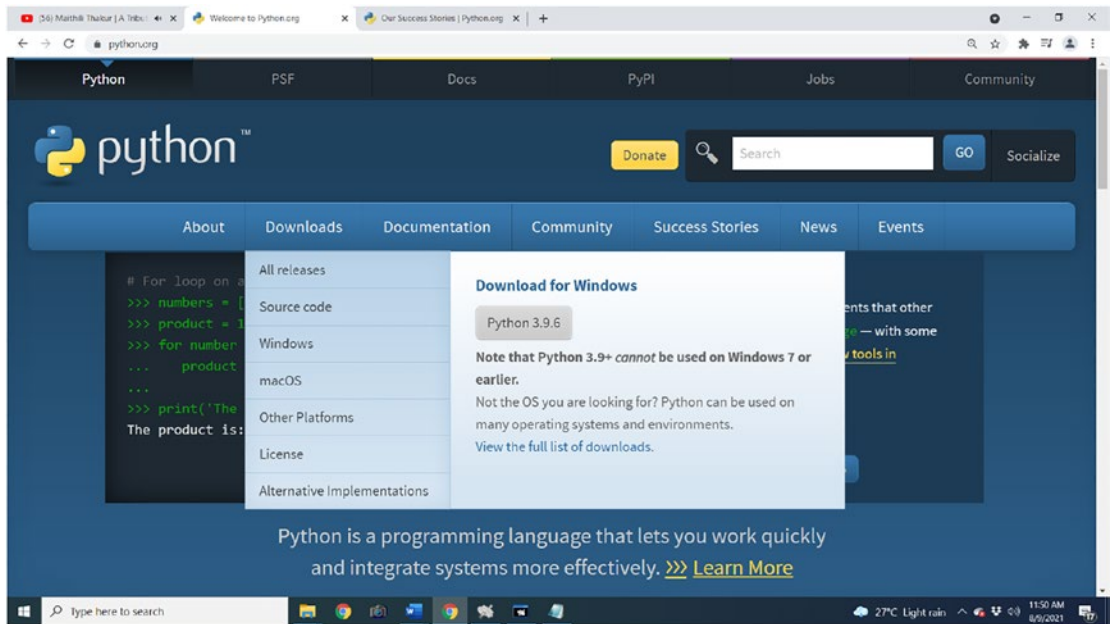


Figure 1-1. Python 3 download for Windows

Once downloaded, open the file. It will show a window shown in Figure 1-2. Do not forget to check all the checkboxes so that the Python's installation folder can be added to the **PATH** variable in the Windows environment. It enables us to launch Python from the command prompt.



Figure 1-2. Python 3 installation for Windows (check all the checkboxes)

Click on the Install Now option (it requires the administrators' privileges). After installation finishes successfully, it shows a message shown in Figure 1-3.

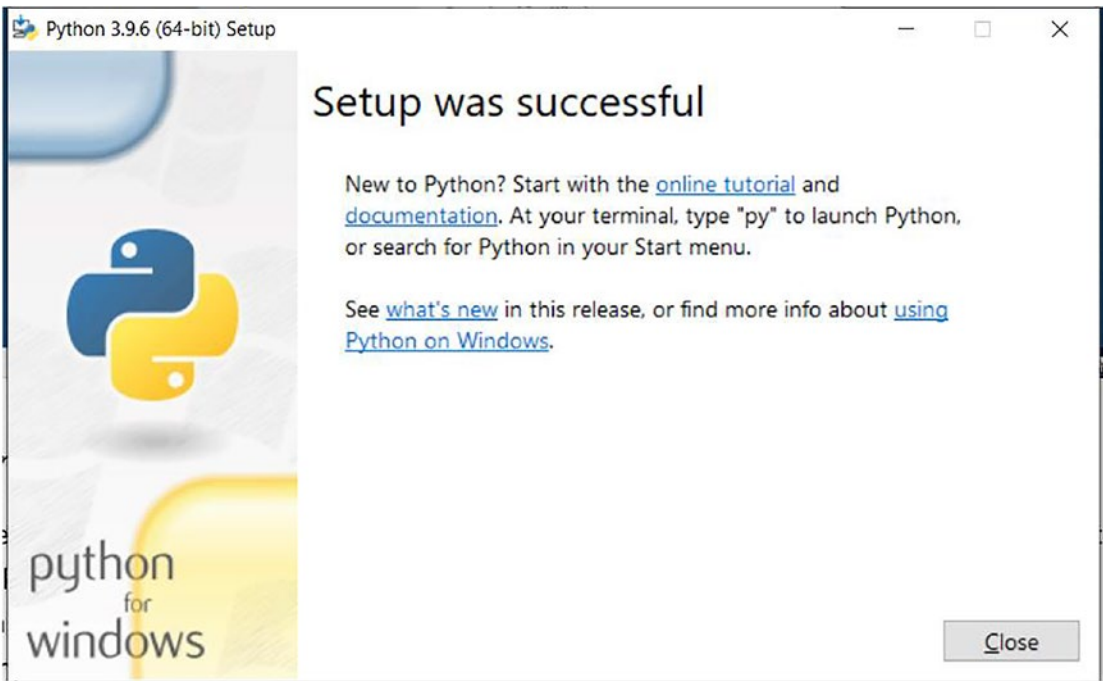


Figure 1-3. Python 3 installation success message

Close the window now, and we are ready for our journey.

Python on Linux Distributions

Python 2 and Python 3 come pre-installed on all the major Linux distributions. We will see that later in this chapter.

Python on macOS

We can get detailed instructions of the installation on macOS at <https://docs.python.org/3/using/mac.html>.

Python Modes

Let us study various Python modes. We will also write our first Python program.

Interactive Mode

Python provides us interactive mode. We can invoke Python interactive mode by opening the IDLE (Integrated Development and Learning Editor) program that comes with the Windows installation. Just type the word IDLE in the Windows search bar and click the IDLE icon that appears as shown in Figure 1-4.

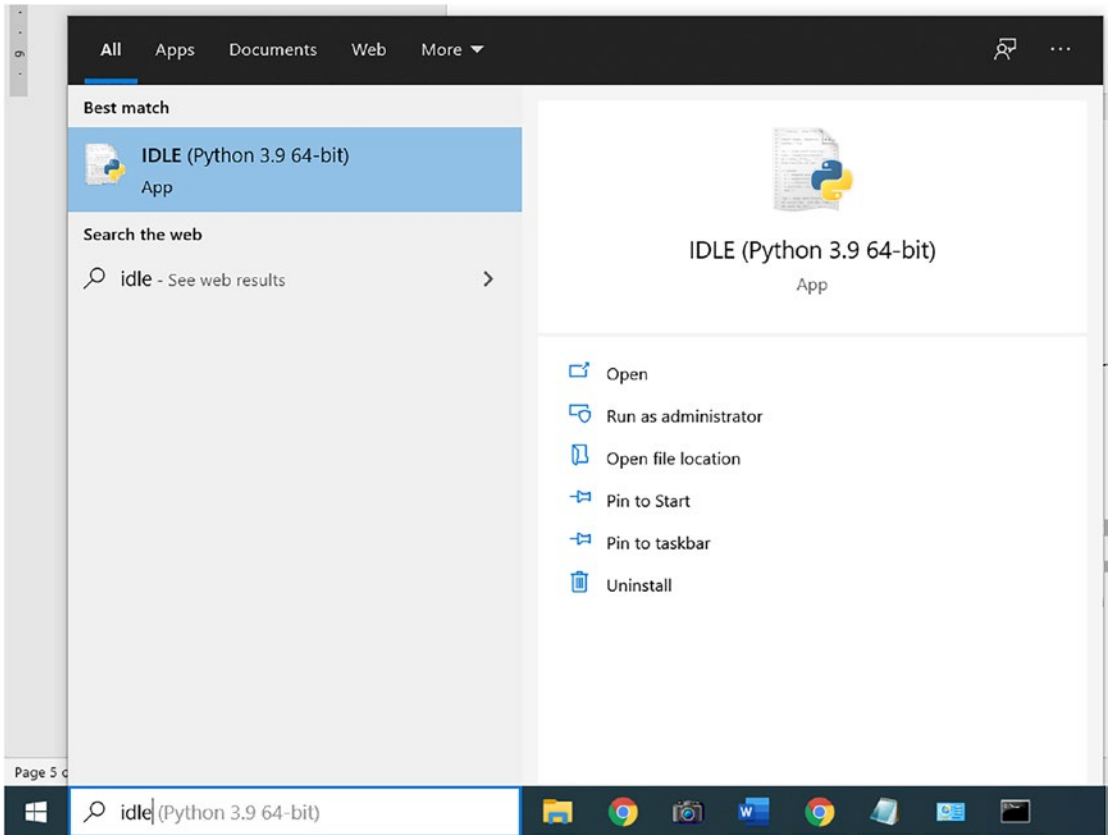


Figure 1-4. *Launching IDLE on Windows*

It will show a window as shown in Figure 1-5.

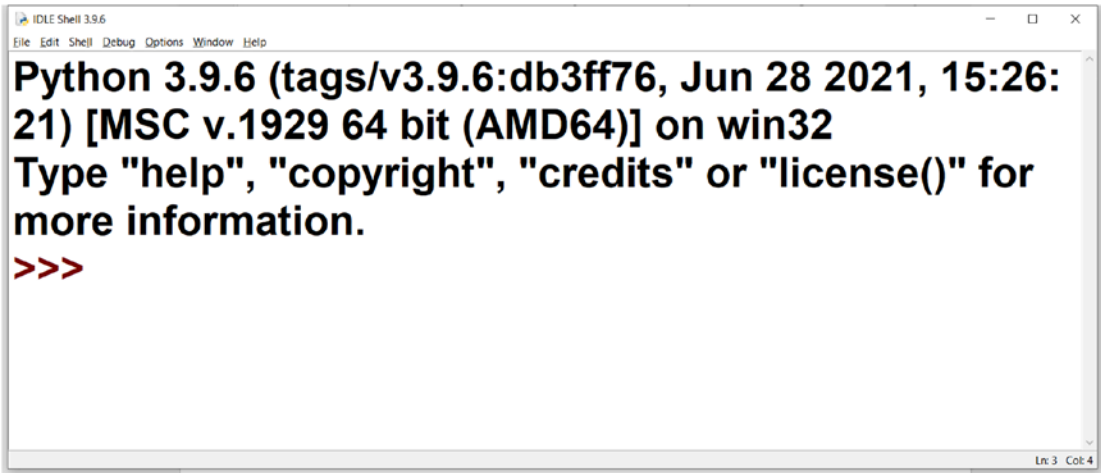


Figure 1-5. IDLE interactive prompt on Windows

We can type in the following code on that:

```
print("Hello, World!")
```

Then hit the enter key to run that. It will show the output as shown in Figure 1-6.

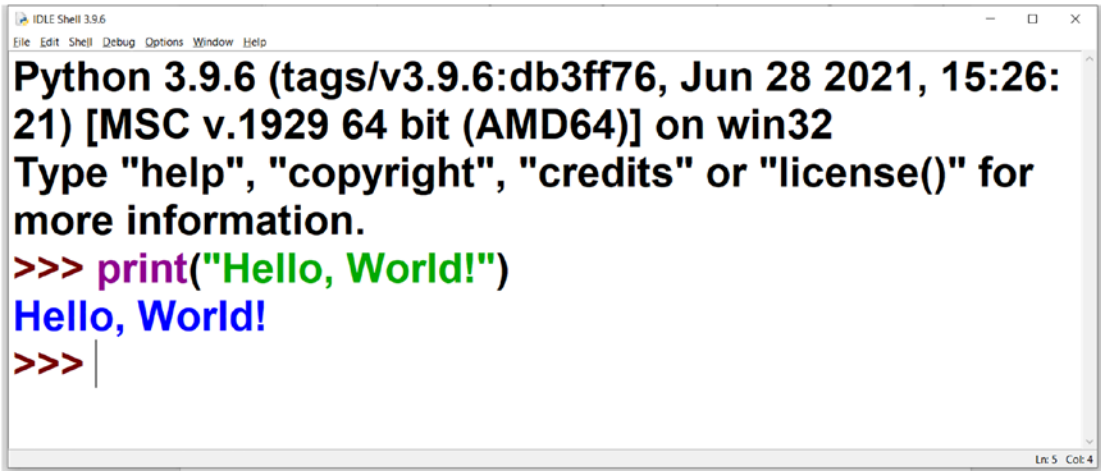
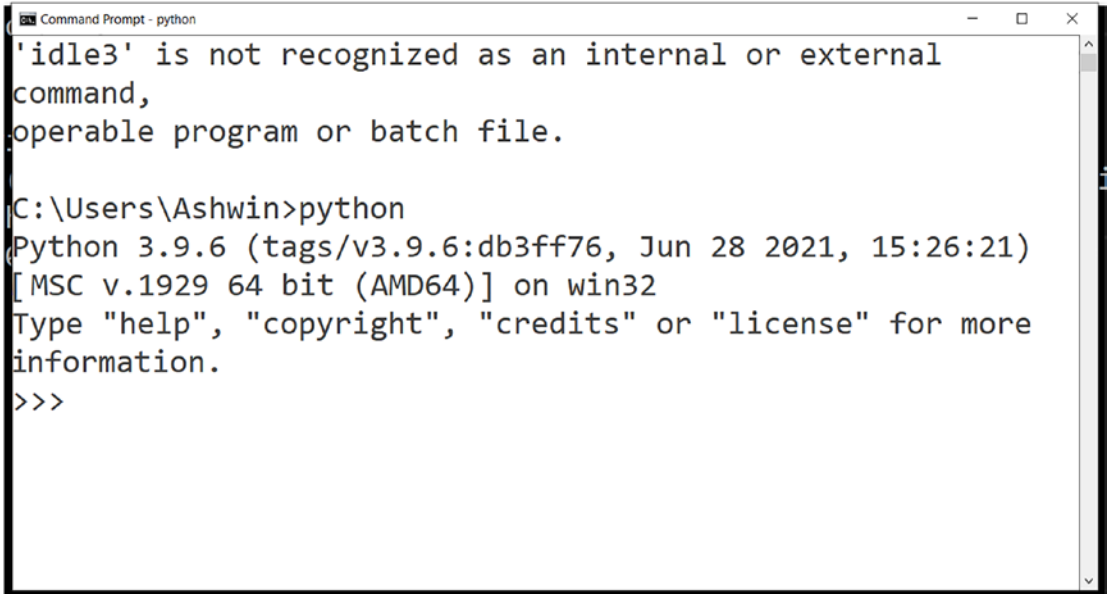


Figure 1-6. Simple code execution

Now we know that IDLE can execute the code line by line or block by block. It is very convenient to run small independent snippets of code here in the interactive prompt.

We can also invoke the interpreter mode without IDLE from the command line by typing in the command in Windows command prompt. The command is `python`. Run the command, and it will launch the interpreter in the interactive mode shown in Figure 1-7.



```

Command Prompt - python
'idle3' is not recognized as an internal or external
command,
operable program or batch file.

C:\Users\Ashwin>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21)
[MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
>>>

```

Figure 1-7. IDLE on Windows command prompt

I use a flavor of Debian distribution (Raspberry Pi OS) on a Raspberry Pi 4 with 8GB of RAM. IDLE does not come pre-installed, but Python 2 and Python 3 are there in the OS. Run the following command on the command prompt (terminal or using an SSH client) to install IDLE for Python 3:

```
sudo pip3 install idle
```

It will install IDLE for Python 3. We will discuss the **pip** utility later in this chapter. Also, without IDLE too, we can invoke the Python 3 interactive mode by typing in the command `python3` on the command prompt on Linux. Note that as the Linux distribution comes with both Python 2 and Python 3, the command `python` will invoke Python 2 on the command prompt. For the Python 3 interpreter, the command is `python3` on Linux. Figure 1-8 shows a Python 3 session in progress in the SSH terminal accessing the Linux command prompt remotely.



Figure 1-8. Python 3 interpreter on the Linux command prompt (remote SSH access)

Now, we can access the IDLE for Python 3 by typing in the command `idle` on the command prompt invoked in the Linux desktop. If we run this command remotely in the SSH terminal, it returns an error as the remote SSH lacks GUI features. We can run this command on the terminal only invoked from the desktop environment of Linux to invoke the IDLE. It could be directly done or even done in a remote desktop environment like VNC as shown in Figure 1-9.

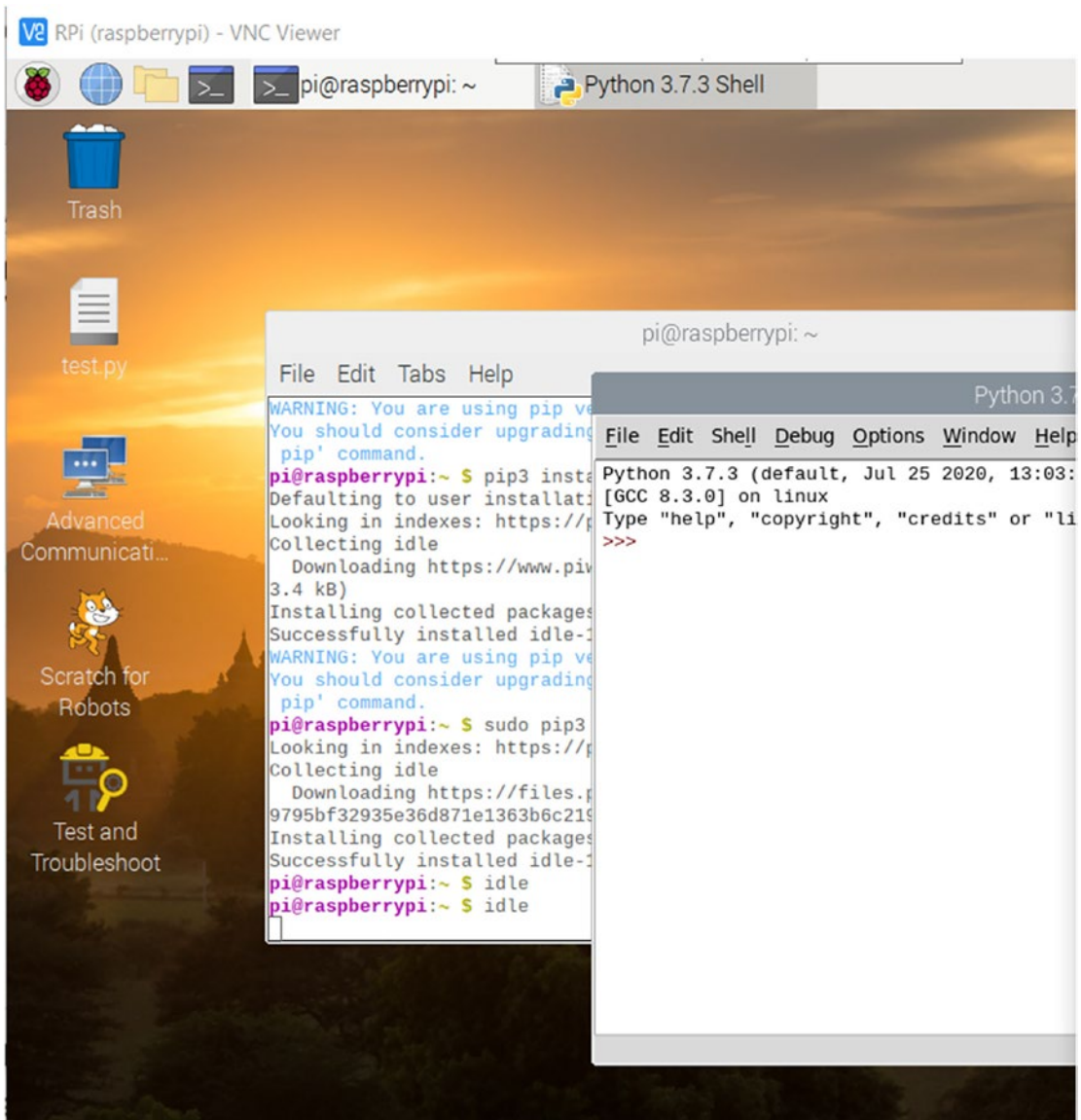


Figure 1-9. Python 3 IDLE invoked on the Linux command prompt (remote Desktop access with VNC)

We can also launch the IDLE from the Raspberry Pi OS menu as shown in Figure 1-10.

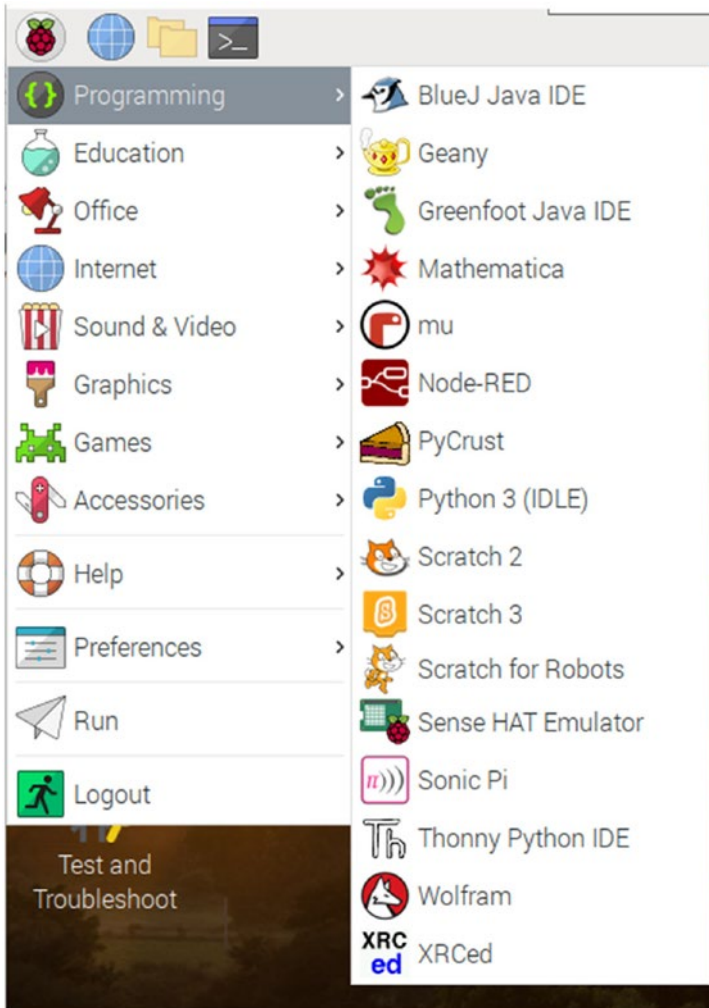


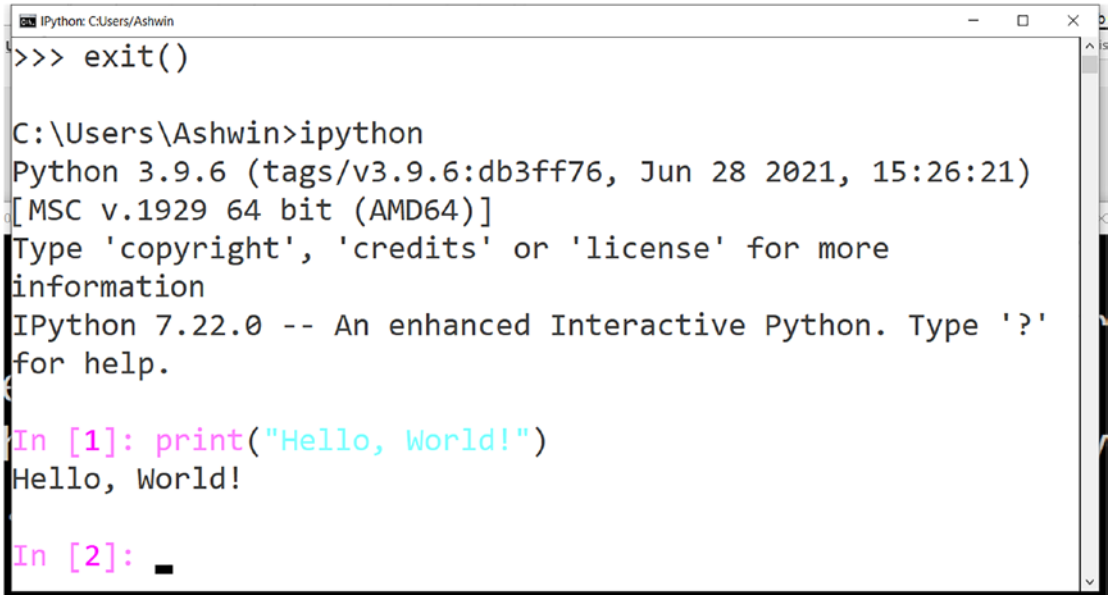
Figure 1-10. IDLE under the Linux menu

Whatever flavor of Linux you are using, you can definitely find the IDLE in the menu and programs once installed. If you do not find it, launch it from the command prompt in the desktop as discussed earlier.

Python 3 also has a more interactive command line environment known as IPython. We can install it by running the following command in the command prompt of Linux and Windows:

```
pip3 install ipython
```

It will install IPython for Python 3 on your OS. We can invoke IPython by typing the command `ipython` in the command prompts of Windows and Linux. In place of IDLE interactive mode, we can also use this. Figure 1-11 shows an IPython session in progress.



```
>>> exit()

C:\Users\Ashwin>ipython
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21)
[MSC v.1929 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more
information
IPython 7.22.0 -- An enhanced Interactive Python. Type '?'
for help.

In [1]: print("Hello, World!")
Hello, World!

In [2]: _
```

Figure 1-11. An IPython session under progress in a Windows command prompt

Finally, the following command terminates the Python interpreter, IDLE interactive mode, and IPython sessions in all the platforms (Windows, macOS, and Linux):

```
exit()
```

This is all about the interactive mode and Python interpreter for now.

Script Mode

Interactive mode of the Python interpreter is easy to get started with and is very good for prototyping and so on. However, for bigger programs, the best way is to use IDLE. Actually, we can use any editor of our choice to write Python programs. However, plaintext editors like Notepad or gedit cannot run the programs. So we use IDEs like IDLE. IDLE is easier to use. Just click the **New File** option in the **File** menu in the menu