

# Deep Learning for Data Architects

---

*Unleash the power of  
Python's deep learning algorithms*

---

**Shekhar Khandelwal**



[www.bpbonline.com](http://www.bpbonline.com)

Copyright © 2024 BPB Online

*All rights reserved.* No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2024

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

**UK | UAE | INDIA | SINGAPORE**

ISBN 978-93-55515-391

[www.bpbonline.com](http://www.bpbonline.com)

**Dedicated to**

*My beloved Wife, Niharika*

*&*

*My Daughter, Saesha*

## About the Author



**Shekhar Khandelwal** is a distinguished Senior AI & Data Scientist, residing in the bustling harbor city of Hamburg, Germany. His academic career shines bright with a Master's degree in Data Science, achieving distinction for his thesis work in the realm of Computer Vision. His name can be spotted in top-tier research papers and publications, predominantly in the area of Deep Learning.

Bringing to the table over 15 years of experience, the author has an extensive professional background in the field of AI and machine learning. His journey ranges from coding and crafting enterprise-level AI products to leading data teams and mentoring future data science professionals. He has successfully developed numerous client solutions utilizing big cloud service platforms such as AWS, Google Cloud, Microsoft Azure, and IBM Cloud.

Despite his deep involvement in the tech industry, our author is also a fitness enthusiast. When he's not making machines smarter, he's likely to be found flexing his muscles at the gym, attending a crossfit class, cycling, or participating in marathon runs. His zeal for fitness is as strong as his passion for AI, making him a well-rounded professional in all respects.



---

## About the Reviewer

**Abonia Sojasingarayar** is a Machine Learning Scientist, Data Scientist, NLP Engineer, Computer Vision Engineer, AI Analyst, Technical Writer, and Technical Book Reviewer with over 6 years of experience. In her 6+ years of professional experience, she has dealt with ML domains (predictive analysis, computer vision, NLP), data analysis, data mining, data visualization, problem-solving, decision-making, planning, and software development. She is skilled in solving modern problems using AI and deep learning using familiar frameworks in Python such as TensorFlow, scikit-learn, Keras, etc., open-source toolkits such as openCV and NLTK, etc., and familiar with Web Marketing Strategies, front-end and back-end development tools, and languages. (For more info, don't hesitate to visit the project section)

She has experience in management consulting, commercial industry, cyber security, RH, chemical industry, banking and insurance, public transport industry, telecommunication services, e-commerce, and information technology sectors.

Her skills and expertise includes studying and transforming data science prototypes, designing machine learning systems, research and implement appropriate ML algorithms and tools, develop machine learning applications according to requirements, select appropriate datasets and data representation methods, run machine learning tests and experiments, perform statistical analysis and fine-tuning using test results, train and retrain systems when necessary, extend existing ML libraries and frameworks, execute analytical experiments to help solve various problems creating an impact across various domains and industries.

## Acknowledgements

First and foremost, this book is dedicated to my uncle, Dinesh Khandelwal. His unwavering faith in me serves as an enduring source of inspiration. A renowned book-seller, his counter is the most honorable stage for my books, each sale symbolizing a cherished victory.

This work is also a tribute to my parents, the architects of my existence, whose love and nurturing have shaped me into the individual I am today.

Lastly, the dedication extends to the two sparkling gems of my life, my wife and daughter. Their smiles, a profound treasure, are the driving forces that fuel my passion.

Special gratitude is extended to BPB Publications for their mentorship and expertise in bringing this work to life. The journey to finalize this book was an enriching process, involving the valuable engagement and collaboration of reviewers, technical experts, and editors.

Acknowledgement is also due to my colleagues and coworkers in the tech industry. Over the years, their wisdom and feedback have been instrumental in my growth and the refinement of my work.

Lastly, my heartfelt thanks go out to every reader who has taken an interest in this book. Your support has been indispensable in transforming this dream into reality. Your enthusiasm fuels my commitment to share my knowledge.

## Preface

The world as we know it is undergoing a profound transformation, a digital metamorphosis that is driven by the immense power of data and artificial intelligence. In the center of this revolutionary tide stands a key player - the data architect. Tasked with the responsibility to make sense of vast data oceans and convert it into meaningful insights, the role of a data architect is evolving at a staggering pace. “Deep Learning for Data Architects” is an embodiment of that evolution.

This book aims to illuminate the path for data architects and enthusiasts, guiding them through the labyrinth of modern data structures and artificial intelligence techniques, from the foundations of Python programming to the advanced landscapes of deep learning models. It is a testament to the compelling narrative of change that the industry is experiencing, catering to professionals who aspire to stay at the forefront of this digital transformation.

“Deep Learning for Data Architects” does not simply regurgitate theoretical concepts. Instead, it creates an engaging dialogue with the reader, providing practical Python implementations for complex AI paradigms, creating a bridge between theory and practice. Each chapter builds upon the last, starting with the basics and gradually delving into the deep end of AI and machine learning.

The journey is as important as the destination, and throughout the course of this book, you will encounter real-world data challenges, explore the depths of neural networks, understand the intricacies of convolutional, recurrent neural networks and unravel the mysteries of Generative Adversarial Networks and Transformers. Each step you take will empower you with new insights and skills, enabling you to tackle any challenge that the data landscape might throw your way.

Whether you are a seasoned data architect aiming to add deep learning to your arsenal, or a budding enthusiast stepping into the exciting intersection of these fields, this book is designed for you. As you turn the pages, you will find yourself not just learning, but evolving with the narrative of deep learning, setting the stage for a future-proof career in this dynamic domain.

Embrace this journey of learning and transformation, and let “Deep Learning for Data Architects” be your guide and companion in the exciting odyssey of AI and data science.

**Chapter 1: Python for Data Science** - serves as a solid foundation, providing an introduction to Python for data science. You will learn essential programming concepts, data structures, and libraries such as NumPy and Pandas. This chapter ensures that you are equipped with the necessary Python skills for the deep learning journey ahead.

**Chapter 2: Real-World Challenges for Data Professionals in Converting Data Into Insights** - dives into the challenges faced by data professionals when converting raw data into valuable insights. You will explore data cleaning, handling missing values, outlier detection, and feature engineering techniques. This chapter prepares you for the data preprocessing steps crucial for successful deep learning implementations.

**Chapter 3: Build a Neural Network-Based Predictive Model** - focuses on building predictive models using neural networks. You will learn about the architecture of a neural network, the role of activation functions, and techniques to handle classification and regression tasks. Through Python code implementations, you will gain hands-on experience in building and training neural networks.

**Chapter 4: Convolutional Neural Networks** - introduces CNNs, a powerful class of neural networks for image analysis tasks. You will understand the key components of CNNs, such as convolutional layers, pooling layers, and fully connected layers. The chapter provides Python code implementations to build CNN models for image classification tasks.

**Chapter 5: Optical Character Recognition** - explores the exciting field of OCR using deep learning. You will discover techniques to extract text from images, enabling automated text recognition. Through Python code implementations, you will learn how to build OCR models and apply them to real-world scenarios.

**Chapter 6: Object Detection** - focuses on object detection, an essential task in computer vision. You will explore popular object detection algorithms and architectures, such as Faster R-CNN and YOLO. Through Python code implementations, you will gain hands-on experience in training and deploying object detection models.

**Chapter 7: Image Segmentation** - delves into image segmentation, a technique used to partition images into meaningful regions. You will learn about popular segmentation algorithms, including U-Net and Mask R-CNN. Through Python code implementations, you will develop a deeper understanding of image segmentation and its applications.

**Chapter 8: Recurrent Neural Networks** - introduces RNNs, which are widely used for sequential data analysis. You will understand the architecture of RNNs, including LSTM and GRU units. Through Python code implementations, you will learn how to build RNN models for tasks such as natural language processing and time series forecasting.

**Chapter 9: Generative Adversarial Networks** - explores the fascinating world of GANs, which can generate new data instances based on training data. You will learn about the adversarial training process and different GAN architectures, including DCGAN and CycleGAN. Through Python code implementations, you will gain hands-on experience in generating realistic images and exploring generative modeling.

**Chapter 10: Transformers** - introduces Transformers, a revolutionary deep learning architecture that has gained prominence in natural language processing tasks. You will learn about the transformer architecture and its variants, such as BERT and GPT. Through Python code implementations, you will gain practical experience in applying transformers to text-related tasks.

## Code Bundle and Coloured Images

Please follow the link to download the *Code Bundle* and the *Coloured Images* of the book:

**<https://rebrand.ly/k8vs91n>**

The code bundle for the book is also hosted on GitHub at **<https://github.com/bpbpublications/Deep-Learning-for-Data-Architects>**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

## Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

**[errata@bpbonline.com](mailto:errata@bpbonline.com)**

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.bpbonline.com](http://www.bpbonline.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

**[business@bpbonline.com](mailto:business@bpbonline.com)** for more details.

At **[www.bpbonline.com](http://www.bpbonline.com)**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

### Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

### If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

### Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

## Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



## Table of Contents

<b>1. Python for Data Science .....</b>	<b>1</b>
Structure .....	2
Objectives .....	2
Setting up the development environment.....	2
<i>Installing Anaconda.....</i>	2
Advance Python libraries for data science.....	8
<i>Numpy.....</i>	8
<i>Pandas .....</i>	10
Reading and writing data to and from different file formats.....	10
<i>Format - csv.....</i>	11
<i>Format - Excel .....</i>	12
<i>Format - JSON .....</i>	12
<i>Format - clipboard .....</i>	13
<i>Format - HTML tables .....</i>	13
<i>Format - PDF .....</i>	15
<i>Format - Web scraping.....</i>	16
Improving efficiency with the pandas read_csv method.....	17
<i>Parameter - dtype .....</i>	19
<i>Parameter - usecols.....</i>	20
<i>Parameter - chunksize .....</i>	22
Conclusion .....	23
Questions.....	23
Answers.....	24
<b>2. Real-World Challenges for Data Professionals in</b>	
<b>Converting Data Into Insights .....</b>	<b>25</b>
Structure .....	26
Objectives .....	26
Pandas profiling .....	26
<i>Analyzing Pandas profile report.....</i>	27
<i>Saving the Pandas profile report to a HTML file .....</i>	32
<i>Creating a Jupyter Notebook widget .....</i>	32
<i>Pandas profile report for big datasets .....</i>	34



Sweetviz .....	34
<i>Installing and getting started with Sweetviz</i> .....	35
<i>Analyzing Sweetviz report</i> .....	35
<i>Generating a report to compare two DataFrames using Sweetviz</i> .....	37
AutoViz .....	38
<i>Installing and getting started with AutoViz</i> .....	38
<i>Analyzing AutoViz report</i> .....	39
Lux 41	
<i>Installing and getting started with Lux</i> .....	41
<i>Analyzing Lux report</i> .....	42
<i>Generating Lux visualizations based on intent</i> .....	43
<i>Saving Lux report to a HTML file</i> .....	44
<i>Advanced features in Lux reports</i> .....	45
Lazy Predict .....	46
<i>Analyzing Lazy Predict experimentation results</i> .....	47
PyCaret .....	48
<i>Installing and getting started with PyCaret</i> .....	48
<i>Analyzing PyCaret experimentation results</i> .....	50
<i>Advanced features of PyCaret</i> .....	51
Conclusion .....	54
Questions .....	54
Answers .....	55
<b>3. Build a Neural Network-Based Predictive Model .....</b>	<b>57</b>
Structure .....	58
Objectives .....	58
Artificial neural network and its components .....	58
<i>Neurons</i> .....	60
<i>Feed forward</i> .....	63
<i>Activation functions</i> .....	63
<i>Loss function</i> .....	64
<i>Backward propagation</i> .....	64
<i>Epoch</i> .....	65
<i>Batch</i> .....	65
<i>Iteration</i> .....	65

Optimizer .....	65
Learning rate .....	66
Building a classification model using neural network .....	66
Problem statement .....	67
Dataset .....	67
Implementation .....	67
Load Python libraries .....	67
Load data .....	67
Descriptive analytics .....	68
Data pre-processing .....	69
Modeling .....	70
Experiment 1 - Hidden layer -1, epoch-100 – shallow neural network .....	70
Experiment 2 - hidden layer -2, epoch-100 – deep neural network .....	75
Building a regression model using neural network .....	78
Problem statement .....	79
Dataset .....	79
Data pre-processing .....	80
Modeling .....	83
Model evaluation .....	85
Conclusion .....	87
Questions .....	88
Answers .....	89
<b>4. Convolutional Neural Networks .....</b>	<b>91</b>
Structure .....	92
Objectives .....	92
Convolutional neural networks components .....	92
Load required libraries .....	92
Digital image as a numpy array .....	92
Kernels/filters and convolution process .....	94
Stride .....	98
Padding .....	99
Convolution on RGB image .....	100
Convolution operation with multiple filters .....	100
One convolution layer .....	101

<i>Pooling</i> .....	102
<i>Flattening</i> .....	103
<i>Dense layers</i> .....	103
Image classification using CNN .....	104
<i>Problem statement</i> .....	104
<i>Dataset - MNIST</i> .....	104
<i>Implementation</i> .....	105
<i>Data pre-processing</i> .....	106
<i>Modeling</i> .....	107
<i>Plot confusion matrix</i> .....	111
<i>Create a confusion matrix and plot</i> .....	113
Hyperparameters tuning using KerasTuner .....	114
<i>Dataset – Fashion MNIST</i> .....	115
<i>Implementation</i> .....	115
<i>Install KerasTuner</i> .....	119
Conclusion .....	125
Questions.....	125
Answers.....	126
<b>5. Optical Character Recognition.....</b>	<b>127</b>
Introduction .....	127
Structure .....	128
Objectives .....	128
Optical character recognition.....	128
OCR Python libraries and their implementation .....	129
<i>Tesseract OCR</i> .....	129
<i>Tesseract OCR demo</i> .....	130
<i>keras-ocr</i> .....	131
<i>keras-ocr demo</i> .....	131
<i>EasyOCR</i> .....	135
<i>EasyOCR demo</i> .....	135
<i>TrOCR</i> .....	140
<i>TrOCR demo</i> .....	141
Conclusion .....	142
Questions.....	143
Answers.....	144

<b>6. Object Detection .....</b>	<b>145</b>
Structure .....	145
Objectives .....	146
Object localization and detection.....	146
Object detection algorithms and their comparison.....	147
<i>Single shot detector Python implementation .....</i>	<i>147</i>
<i>YOLO v3 Python implementation .....</i>	<i>153</i>
<i>Experiment 2 .....</i>	<i>156</i>
<i>Experiment 3 .....</i>	<i>156</i>
<i>Experiment 4 .....</i>	<i>157</i>
<i>Experiment 5 .....</i>	<i>158</i>
Conclusion .....	158
Questions.....	159
Answers.....	159
<b>7. Image Segmentation .....</b>	<b>161</b>
Structure .....	161
Objectives .....	162
Difference between image classification, detection and segmentation.....	162
Image segmentation architectures.....	163
U-Net Python implementation .....	163
FCN-8 Python implementation.....	170
Mask R-CNN Python implementation .....	175
Conclusion .....	183
Questions.....	184
Answers.....	185
<b>8. Recurrent Neural Networks .....</b>	<b>187</b>
Structure .....	188
Objectives .....	188
Algorithms for RNN implementation.....	188
RNN implementation.....	189
Long short-term memory implementation .....	193
Gated Recurrent Unit implementation .....	197
Conclusion .....	200
Multiple choice questions.....	200
Answers.....	201

<b>9. Generative Adversarial Networks.....</b>	<b>203</b>
Structure .....	204
Objectives .....	204
Types of GAN .....	204
Vanilla GAN Python implementation.....	205
Key difference between Vanilla GAN and DCGAN .....	210
DCGAN Python implementation .....	212
StyleGAN Python implementation .....	218
<i>Setup environment</i> .....	218
Conclusion .....	222
Questions.....	223
Answers.....	224
<b>10. Transformers.....</b>	<b>225</b>
Structure .....	226
Objectives .....	226
Introduction to Transformers in deep learning .....	226
Various transformers architectures.....	227
Difference between contextual and non-contextual embeddings.....	228
BERT Python implementation.....	229
GPT Python implementation.....	232
Conclusion .....	235
Questions.....	235
Answers.....	236
 <b>Index.....</b>	 <b>237-243</b>

## **Join our book's Discord space**

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

**<https://discord.bpbonline.com>**



# CHAPTER 1

# Python for Data Science

*You can't build a great building on a weak foundation. You must have a solid foundation if you are going to have a super-strong structure.*

— *Gordon B. Hinkley*

Data is the most important component of data science. Python libraries for data science are built specifically to solve various peripheral issues that Data Scientists may face, like data sourcing, cleaning, pre-processing, and working with big data. It is important to know what these libraries have to offer and the sneaky tricks that can be implemented in our day-to-day data wrangling. The high-level methods within these libraries save you a lot of time performing humongous tasks, using just a few lines of code.

Additionally, data has to be sourced from all over the place. Hence, as someone who is building predictive models, maybe in academia or at work, should be familiar with different kinds of data that one can deal with and how to source them using Python.

Plus, when you work on your personal system, you have to be really cautious about how you utilize the available resources in terms of computing hardware and so on. Hence, it is of utmost importance to know various tricks that can help you with memory management. The tricks that usually go unnoticed and unexplored

within Python packages, if explored, can significantly improve your efficiency while dealing with data.

## Structure

In this chapter, we will cover the following topics:

- Setting up the development environment
- Advance Python libraries for data science
- Reading and writing data to and from various file formats
- Improving efficiency with the pandas `read_csv` method

## Objectives

After studying this chapter, you should be able to set up your laptop with the required tools and technologies to embark on your journey of implementing various deep learning models that will be discussed in this book. Also, you should be able to install various Python packages that are prerequisites for building any predictive model. Once the development environment is set up, you will learn about various Python libraries that are available for scientific computing, machine learning and deep learning. Additionally, you will learn how to maximize your productivity in a limited hardware environment, since machine learning and deep learning are computationally expensive.

## Setting up the development environment

As part of the environment setup, we will look at how to install the Anaconda tool, which gives access to almost all the required underlying tools and technologies for development. Further, we will understand how to get started with Jupyter Notebook, which is the IDE for Data Scientists. Finally, we will look at how to enhance the notebook by installing important plugins for better usability of the tool.

## Installing Anaconda

The first and foremost tool that has to be installed on your machine is Anaconda. Anaconda is a distribution of packages for the Python and R programming languages. It includes a package manager called conda that can be used to install, update, and manage packages within the Anaconda environment.



Tools that come along with Anaconda are as follows:

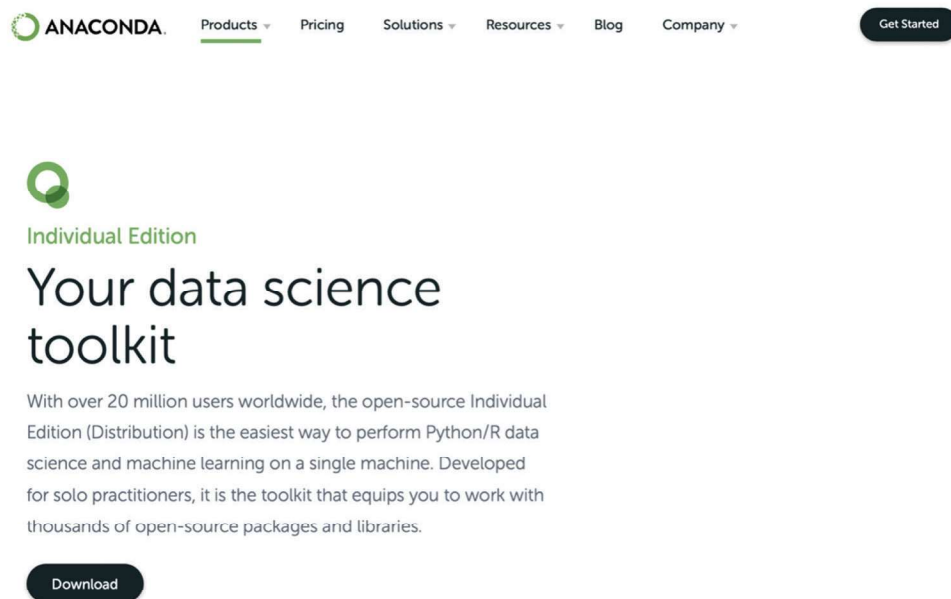
- Jupyter Notebook
- Orange
- Spyder
- PyCharm
- VS Code
- RStudio
- IBM Watson Studio

Each of these software can be installed separately as well. However, with anaconda, all of them can be installed in one shot.

Follow the steps to install Anaconda on your machine:

1. Go to the home page of Anaconda at <https://www.anaconda.com/products/individual>.

It will look like as shown in *Figure 1.1*:



*Figure 1.1: Anaconda home page*

2. Click on **Download** and download the installer of the choice of your operating system, as shown in the following screenshot:

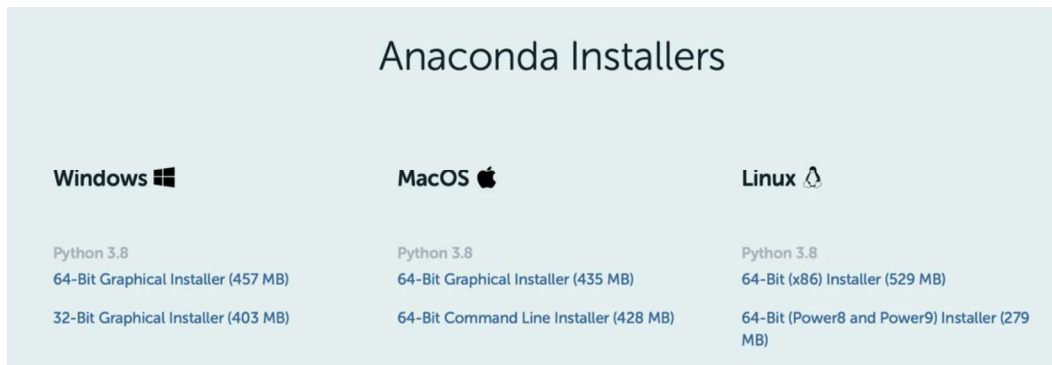


Figure 1.2: Anaconda installers

3. Double-click on the downloaded installer and follow the installation wizard, as shown in the following screenshot:

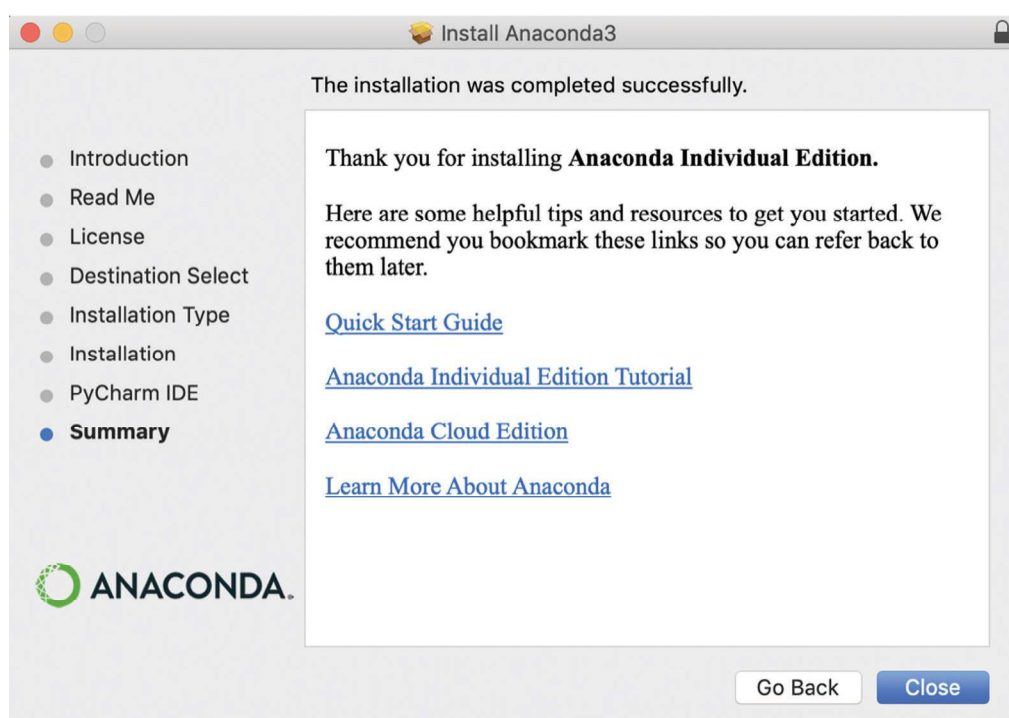


Figure 1.3: Anaconda installation wizard

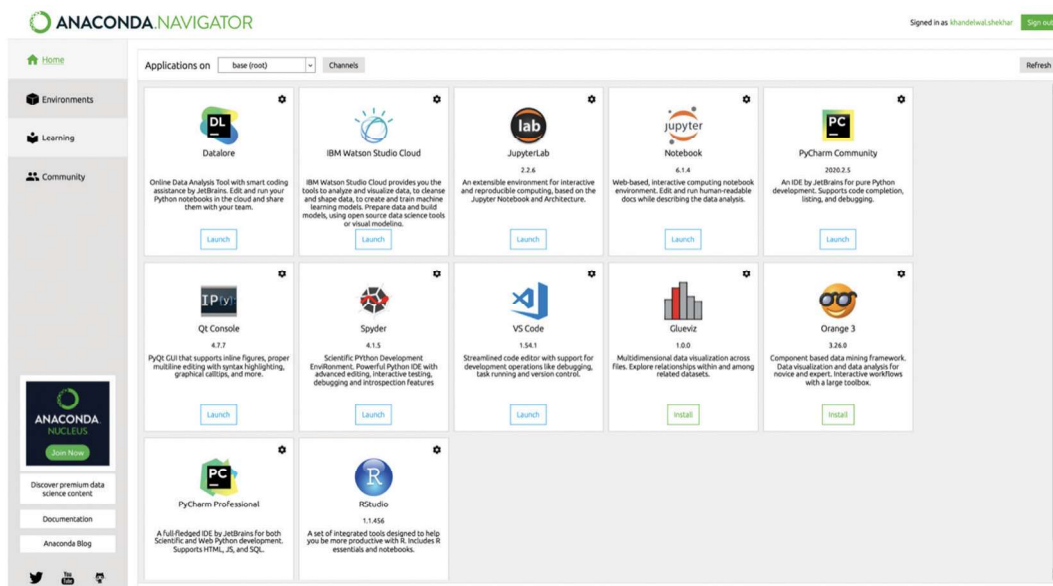
4. Click the **Close** button after you complete the installation of Anaconda on your machine.

Now, let's get started with Jupyter Notebook.

## Getting Started with Jupyter Notebook

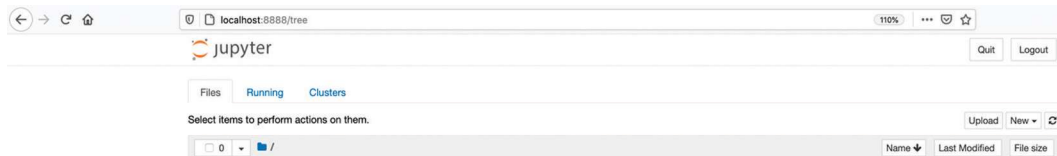
Once Anaconda is installed, you can proceed to launch the Jupyter Notebook on the web browser:

1. Open the **Anaconda Navigator** app on your machine. The screen will look as shown in *Figure 1.4*:



*Figure 1.4: Anaconda Navigator*

2. Click on the **Launch** button under the Jupyter Notebook icon. This will launch the Jupyter homepage on your default browser, as shown in the following screenshot:



*Figure 1.5: Jupyter Notebook*

Further, you can navigate to the desired folder in your filesystem and go to the folder where you want to create a notebook to start building your projects.

3. Click on the **New** dropdown and click on the **Python 3** option, as shown in the following screenshot:

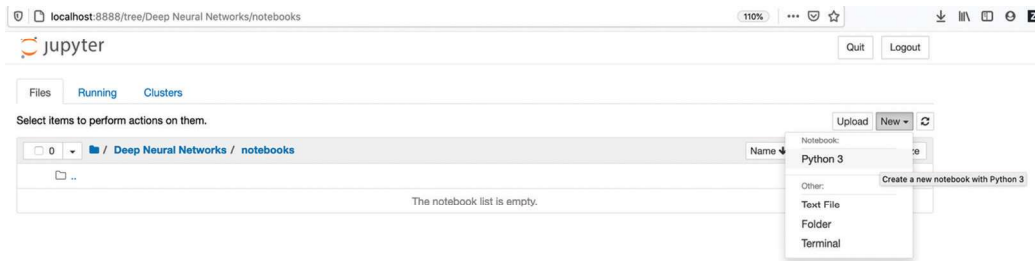


Figure 1.6: Notebook IDE

4. This will create a new notebook with the name **Untitled**, which can be renamed as shown in the following screenshot:

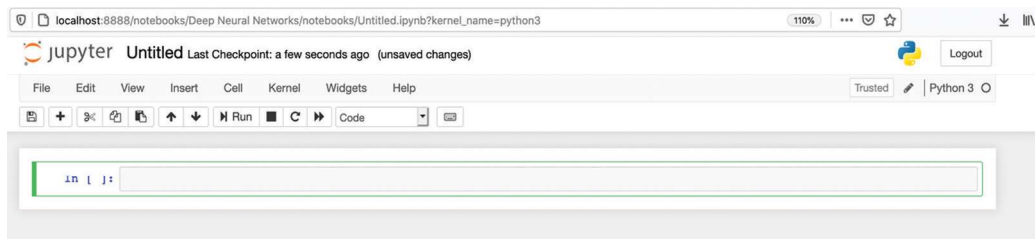


Figure 1.7: Notebook cell

5. You can start writing Python code in the cells, as shown in Figure 1.6, and run them right there using the respective buttons provided at the top of the IDE:

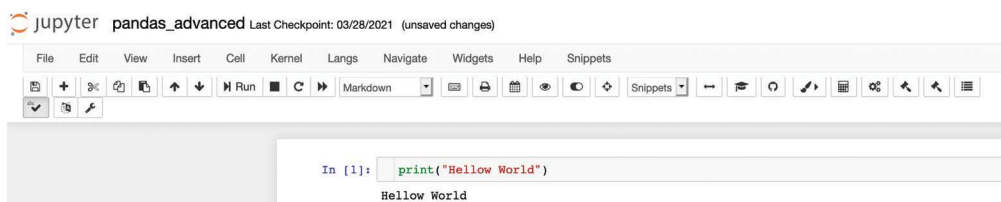


Figure 1.8: Notebook Hello World

## Enhancing Jupyter Notebook usability

Many seasoned data scientists usually work on the vanilla setup of Jupyter notebook. However, it is important to know that there are many more amazing features of the Jupyter Notebook that are usually not explored by most users. Once you explore them and start using them, you will realize how important they are to further enhance your usability and productivity with building models and applications using Jupyter Notebook.

These features are the ability to incorporate latex in the markdown cells of the notebook, code prettify for making the code look neater to promote best practices and preserve its integrity, the ability to automatically create a table of contents of your exploratory data analysis steps, code auto-completion with notebook, the ability to save code snippets for reusability, and many more.

## Installing the required Python packages

Python, as we know, is a multi-utility programming language. It is also the most preferred programming language for data science due to the amazing scientific computing, visualization and algorithm implementation libraries.

In this section, we will install the most important Python packages that are required to build any machine learning and deep learning model in Python.

Those are as follows:

- Pandas
- NumPy
- Matplotlib
- Seaborn
- Scikit Learn
- TensorFlow
- Keras
- Scikit-image
- OpenCV

Use *Code 1.1* snippet and run them on the notebook cell for installing the Python packages on your machine. These are simple pip install commands that can be run on the command line too. In order to run them through the notebook cell, add an `“!”` before the command. That will make the notebook know that the code has to be interpreted and executed as a shell command:

1. `import sys`
2. `!{sys.executable} -m pip install -U pandas`
3. `!{sys.executable} -m pip install -U numpy`
4. `!{sys.executable} -m pip install -U matplotlib`
5. `!{sys.executable} -m pip install -U seaborn`
6. `!{sys.executable} -m pip install -U scikit-learn`
7. `!{sys.executable} -m pip install -U tensorflow`

```
8. !{sys.executable} -m pip install -U keras
9. !{sys.executable} -m pip install -U scikit-image
10. !{sys.executable} -m pip install -U opencv-python
```

*Code 1.1: pip install commands*

Note that these are some basic Python libraries required for building any machine learning or deep learning model. However, additional libraries required at specific stages of the model building process will be introduced and discussed in their respective chapters.

## Advance Python libraries for data science

Python has become the most preferred language for data science, owing to the amazing and easy-to-use scientific computing and model building libraries. Libraries like Pandas and Numpy provide various high-level methods for data processing. Libraries like matplotlib and seaborn make data visualization a piece of cake. Libraries like scikit-learn provide high-level methods to implement almost all machine learning algorithms. Libraries like Keras and PyTorch provide high-level methods to implement almost all deep learning algorithms.

Also, Python for data science has a very active online community, which constantly contributes to building various high-level Python packages for easy and holistic implementation of almost all stages of the data science life cycle. This makes data sourcing, data cleaning, data pre-processing, data visualization, and model building possible with just a couple of lines of code implementation.

## Numpy

Numpy is a Python library that is written for scientific computing and data analysis. It stands for **Numerical Python**.

When it comes to data science, the data is usually big data. Even when we work on sample data for building and evaluating a predictive model, the data is huge enough to give your machine a hard time to process. Now, Python as we know it may not be that efficient in terms of computational speed and efficiency. Hence, the power of vectorization that is inherently present in the advanced Python libraries, like Numpy, comes to the rescue for faster computations.

In standard Python ways, when you need to deal with data in an iterative fashion, “for” loops are the best premise. In a loop, records are treated one row at a time, which is time-consuming and not very efficient.