Learning Predictive Analytics with Python

This book is your guide to getting started with predictive analytics using Python. The book strives to maintain a balance between mathematical concepts and details of implementation of the algorithms. You will learn how to process data and make predictive models in Python using libraries such as pandas, scikit-learn, and NumPy.

You will start by getting an understanding of the basics of predictive modelling, followed by a demonstration of methods for cleansing the dataset of impurities and getting it ready for modeling. You will also learn more about the commonly used predictive modeling algorithms such as linear regression, decision trees, clustering, and logistic regression. Finally, you will see the best practices in predictive modelling, as well as the different applications of predictive modeling in the modern world.

Who this book is written for

If you wish to learn how to implement Predictive Analytics algorithms using Python libraries, then this is the book for you. If you are familiar with coding in Python (or some other programming/statistical/scripting language), then this book should also help you. Some familiarity with Python will be useful to get the most out of this book, but it is certainly not a prerequisite.

What you will learn from this book

- Understand the statistical and mathematical concepts behind Predictive Analytics algorithms and implement Predictive Analytics algorithms using Python libraries
- Analyze the result parameters arising from the implementation of Predictive Analytics algorithms
- Write Python modules/functions from scratch to execute segments or the whole of these algorithms
- Recognize and mitigate various contingencies and issues related to the implementation of Predictive Analytics algorithms
- Get to know various methods of importing, cleaning, sub-setting, merging, joining, concatenating, exploring, grouping, and plotting data with pandas and NumPy
- Create dummy datasets and simple mathematical simulations using the NumPy and pandas libraries
- Understand the best practices while handling datasets in Python and creating predictive models out of them


Foreword by Pradeep Gulipalli, Co-founder and Head of India Operations - Tiger Analytics

Ashish Kumar
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Getting started with Predictive Modelling'
- A synopsis of the book’s content
- More information on Learning Predictive Analytics with Python
Ashish Kumar has a B. Tech from IIT Madras and is a Young India Fellow from the batch of 2012-13. He is a data science enthusiast with extensive work experience in the field. As a part of his work experience, he has worked with tools, such as Python, R, and SAS. He has also implemented predictive algorithms to glean actionable insights for clients from transport and logistics, online payment, and healthcare industries. Apart from the data sciences, he is enthused by and adept at financial modelling and operational research. He is a prolific writer and has authored several online articles and short stories apart from running his own analytics blog. He also works pro-bono for a couple of social enterprises and freelances his data science skills.

He can be contacted on LinkedIn at https://goo.gl/yqrfo4, and on Twitter at https://twitter.com/asis64.
Social media and the Internet of Things have resulted in an avalanche of data. The data is powerful but not in its raw form; it needs to be processed and modelled and Python is one of the most robust tools we have out there to do so. It has an array of packages for predictive modelling and a suite of IDEs to choose from. Learning to predict who would win, lose, buy, lie, or die with Python is an indispensable skill set to have in this data age.

This book is your guide to get started with Predictive Analytics using Python as the tool. You will learn how to process data and make predictive models out of them. A balanced weightage has been given to both the statistical and mathematical concepts and implementing them in Python using libraries, such as pandas, scikit-learn, and NumPy. Starting with understanding the basics of predictive modelling, you will see how to cleanse your data of impurities and make it ready for predictive modelling. You will also learn more about the best predictive modelling algorithms, such as linear regression, decision trees, and logistic regression. Finally, you will see what the best practices in predictive modelling are, as well as the different applications of predictive modelling in the modern world.

What this book covers

Chapter 1, Getting Started with Predictive Modelling, talks about aspects, scope, and applications of predictive modelling. It also discusses various Python packages commonly used in data science, Python IDEs, and the methods to install these on systems.

Chapter 2, Data Cleaning, describes the process of reading a dataset, getting a bird's eye view of the dataset, handling the missing values in the dataset, and exploring the dataset with basic plotting using the pandas and matplotlib packages in Python. The data cleaning and wrangling together constitutes around 80% of the modelling time.
Chapter 3, *Data Wrangling*, describes the methods to subset a dataset, concatenate or merge two or more datasets, group the dataset by categorical variables, split the dataset into training and testing sets, generate dummy datasets using random numbers, and create simulations using random numbers.

Chapter 4, *Statistical Concepts for Predictive Modelling*, explains the basic statistics needed to make sense of the model parameters resulting from the predictive models. This chapter deals with concepts like hypothesis testing, z-tests, t-tests, chi-square tests, p-values, and so on followed by a discussion on correlation.

Chapter 5, *Linear Regression with Python*, starts with a discussion on the mathematics behind the linear regression validating the mathematics behind it using a simulated dataset. It is then followed by a summary of implications and interpretations of various model parameters. The chapter also describes methods to implement linear regression using the statsmodel.api and scikit-learn packages and handling various related contingencies, such as multiple regression, multi-collinearity, handling categorical variables, non-linear relationships between predictor and target variables, handling outliers, and so on.

Chapter 6, *Logistic Regression with Python*, explains the concepts, such as odds ratio, conditional probability, and contingency tables leading ultimately to detailed discussion on mathematics behind the logistic regression model (using a code that implements the entire model from scratch) and various tests to check the efficiency of the model. The chapter also describes the methods to implement logistic regression in Python and drawing and understanding an ROC curve.

Chapter 7, *Clustering with Python*, discusses the concepts, such as distances, the distance matrix, and linkage methods to understand the mathematics and logic behind both hierarchical and k-means clustering. The chapter also describes the methods to implement both the types of clustering in Python and methods to fine tune the number of clusters.

Chapter 8, *Trees and Random Forests with Python*, starts with a discussion on topics, such as entropy, information gain, gini index, and so on. To illustrate the mathematics behind creating a decision tree followed by a discussion on methods to handle variations, such as a continuous numerical variable as a predictor variable and handling a missing value. This is followed by methods to implement the decision tree in Python. The chapter also gives a glimpse into understanding and implementing the regression tree and random forests.

Chapter 9, *Best Practices for Predictive Modelling*, entails the best practices to be followed in terms of coding, data handling, algorithms, statistics, and business context for getting good results in predictive modelling.
Appendix, *A List of Links*, contains a list of sources which have been directly or indirectly consulted or used in the book. It also contains the link to the folder which contains datasets used in the book.
Getting Started with Predictive Modelling

Predictive modelling is an art; its a science of unearthing the story impregnated into silos of data. This chapter introduces the scope and application of predictive modelling and shows a glimpse of what could be achieved with it, by giving some real-life examples.

In this chapter, we will cover the following topics in detail:

• Introducing predictive modelling
• Applications and examples of predictive modelling
• Installing and downloading Python and its packages
• Working with different IDEs for Python

Introducing predictive modelling

Did you know that Facebook users around the world share 2,460,000 pieces of content every minute of the day? Did you know that 72-hours worth of new video content is uploaded on YouTube in the same time and, brace yourself, did you know that everyday around 2.5 exabytes (10^18) of data is created by us humans? To give you a perspective on how much data that is, you will need a million 1 TB (1000 GB) hard disk drives every day to store that much data. In a year, we will outgrow the US population and will be north of five times the UK population and this estimation is by assuming the fact that the rate of the data generation will remain the same, which in all likelihoods will not be the case.
The breakneck speed at which the social media and Internet of Things have grown is reflected in the huge silos of data humans generate. The data about where we live, where we come from, what we like, what we buy, how much money we spend, where we travel, and so on. Whenever we interact with a social media or Internet of Things website, we leave a trail, which these websites gleefully log as their data. Every time you buy a book at Amazon, receive a payment through PayPal, write a review on Yelp, post a photo on Instagram, do a check-in on Facebook, apart from making business for these websites, you are creating data for them.

Harvard Business Review (HBR) says "Data is the new oil" and that "Data Scientist is the sexiest job of the 21st century". So, why is the data so important and how can we realize the full potential of it? There are broadly two ways in which the data is used:

- **Retrospective analytics**: This approach helps us analyze history and glean out insights from the data. It allows us to learn from mistakes and adopt best practices. These insights and learnings become the torchbearer for the purpose of devising better strategy. Not surprisingly, many experts have been claiming that data is the new middle manager.

- **Predictive analytics**: This approach unleashes the might of data. In short, this approach allows us to predict the future. Data science algorithms take historical data and spit out a statistical model, which can predict who will buy, cheat, lie, or die in the future.

Let us evaluate the comparisons made with oil in detail:

- Data is as abundant as oil used to be, once upon a time, but in contrast to oil, data is a non-depleting resource. In fact, one can argue that it is reusable, in the sense that, each dataset can be used in more than one way and also multiple number of times.

- It doesn't take years to create data, as it takes for oil.

- Oil in its crude form is worth nothing. It needs to be refined through a comprehensive process to make it usable. There are various grades of this process to suit various needs; it's the same with data. The data sitting in silos is worthless; it needs to be cleaned, manipulated, and modelled to make use of it. Just as we need refineries and people who can operate those refineries, we need tools that can handle data and people who can operate those tools. Some of the tools for the preceding tasks are Python, R, SAS, and so on, and the people who operate these tools are called **data scientists**.
A more detailed comparison of oil and data is provided in the following table:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td><strong>Oil</strong></td>
<td></td>
</tr>
<tr>
<td>It's a non-depleting resource and also reusable.</td>
<td>It's a depleting resource and non-reusable.</td>
<td></td>
</tr>
<tr>
<td>Data collection requires some infrastructure or system in place. Once the system is in place, the data generation happens seamlessly.</td>
<td>Drilling oil requires a lot of infrastructure. Once the infrastructure is in place, one can keep drawing the oil until the stock dries up.</td>
<td></td>
</tr>
<tr>
<td>It needs to be cleaned and modelled.</td>
<td>It needs to be cleaned and processed.</td>
<td></td>
</tr>
<tr>
<td>The time taken to generate data varies from fractions of second to months and years.</td>
<td>It takes decades to generate.</td>
<td></td>
</tr>
<tr>
<td>The worth and marketability of different kinds of data is different.</td>
<td>The worth of crude oil is same everywhere. However, the price and marketability of different end products of refinement is different.</td>
<td></td>
</tr>
<tr>
<td>The time horizon for monetization of data is smaller after getting the data.</td>
<td>The time horizon for monetizing oil is longer than that for data.</td>
<td></td>
</tr>
</tbody>
</table>

**Scope of predictive modelling**

Predictive modelling is an *ensemble of statistical algorithms* coded in a *statistical tool*, which when applied on *historical data*, outputs a *mathematical function* (or equation). It can in-turn be used to predict outcomes based on some inputs (on which the model operates) from the future to drive a goal in business context or enable better decision making in general.

To understand what predictive modelling entails, let us focus on the phrases highlighted previously.

**Ensemble of statistical algorithms**

Statistics are important to understand data. It tells volumes about the data. How is the data distributed? Is it centered with little variance or does it varies widely? Are two of the variables dependent on or independent of each other? Statistics helps us answer these questions. This book will expect a basic understanding of basic statistical terms, such as mean, variance, co-variance, and correlation. Advanced terms, such as hypothesis testing, Chi-Square tests, p-values, and so on will be explained as and when required. Statistics are the cog in the wheel called model.
Getting Started with Predictive Modelling

Algorithms, on the other hand, are the blueprints of a model. They are responsible for creating mathematical equations from the historical data. They analyze the data, quantify the relationship between the variables, and convert it into a mathematical equation. There is a variety of them: Linear Regression, Logistic Regression, Clustering, Decision Trees, Time-Series Modelling, Naïve Bayes Classifiers, Natural Language Processing, and so on. These models can be classified under two classes:

- **Supervised algorithms**: These are the algorithms wherein the historical data has an output variable in addition to the input variables. The model makes use of the output variables from historical data, apart from the input variables. The examples of such algorithms include Linear Regression, Logistic Regression, Decision Trees, and so on.

- **Un-supervised algorithms**: These algorithms work without an output variable in the historical data. The example of such algorithms includes clustering.

The selection of a particular algorithm for a model depends majorly on the kind of data available. The focus of this book would be to explain methods of handling various kinds of data and illustrating the implementation of some of these models.

**Statistical tools**

There are a many statistical tools available today, which are laced with inbuilt methods to run basic statistical chores. The arrival of open-source robust tools like R and Python has made them extremely popular, both in industry and academia alike. Apart from that, Python's packages are well documented; hence, debugging is easier.

Python has a number of libraries, especially for running the statistical, cleaning, and modelling chores. It has emerged as the first among equals when it comes to choosing the tool for the purpose of implementing preventive modelling. As the title suggests, Python will be the choice for this book, as well.

**Historical data**

Our machinery (model) is built and operated on this oil called data. In general, a model is built on the historical data and works on future data. Additionally, a predictive model can be used to fill missing values in historical data by interpolating the model over sparse historical data. In many cases, during modelling stages, future data is not available. Hence, it is a common practice to divide the historical data into training (to act as historical data) and testing (to act as future data) through sampling.
As discussed earlier, the data might or might not have an output variable. However, one thing that it promises to be is messy. It needs to undergo a lot of cleaning and manipulation before it can become of any use for a modelling process.

**Mathematical function**

Most of the data science algorithms have underlying mathematics behind them. In many of the algorithms, such as regression, a mathematical equation (of a certain type) is assumed and the parameters of the equations are derived by fitting the data to the equation.

For example, the goal of linear regression is to fit a linear model to a dataset and find the equation parameters of the following equation:

\[ Y = \alpha_0 + \beta_1X_1 + \beta_2X_2 + \ldots + \beta_nX_n \]

The purpose of modelling is to find the best values for the coefficients. Once these values are known, the previous equation is good to predict the output. The equation above, which can also be thought of as a linear function of Xi’s (or the input variables), is the linear regression model.

Another example is of logistic regression. There also we have a mathematical equation or a function of input variables, with some differences. The defining equation for logistic regression is as follows:

\[
P = \frac{e^{a+bx}}{1 + e^{a+bx}} = \frac{1}{1 + e^{-(a+bx)}}
\]

Here, the goal is to estimate the values of \(a\) and \(b\) by fitting the data to this equation. Any supervised algorithm will have an equation or function similar to that of the model above. For unsupervised algorithms, an underlying mathematical function or criterion (which can be formulated as a function or equation) serves the purpose. The mathematical equation or function is the backbone of a model.

**Business context**

All the effort that goes into predictive analytics and all its worth, which accrues to data, is because it solves a business problem. A business problem can be anything and it will become more evident in the following examples:

- Trickling the users of the product/service to buy more from you by increasing the click through rates of the online ads
• Predicting the probable crime scenes in order to prevent them by aggregating an invincible lineup for a sports league
• Predicting the failure rates and associated costs of machinery components
• Managing the churn rate of the customers

The predictive analytics is being used in an array of industries to solve business problems. Some of these industries are, as follows:

• Banking
• Social media
• Retail
• Transport
• Healthcare
• Policing
• Education
• Travel and logistics
• E-commerce
• Human resource

By what quantum did the proposed solution make life better for the business, is all that matters. That is the reason; predictive analytics is becoming an indispensable practice for management consulting.

In short, predictive analytics sits at the sweet spot where statistics, algorithm, technology and business sense intersect. Think about it, a mathematician, a programmer, and a business person rolled in one.

**Knowledge matrix for predictive modelling**

As discussed earlier, predictive modelling is an interdisciplinary field sitting at the interface and requiring knowledge of four disciplines, such as Statistics, Algorithms, Tools, Techniques, and Business Sense. Each of these disciplines is equally indispensable to perform a successful task of predictive modelling.

These four disciplines of predictive modelling carry equal weights and can be better represented as a knowledge matrix; it is a symmetric 2 x 2 matrix containing four equal-sized squares, each representing a discipline.
Task matrix for predictive modelling

The tasks involved in predictive modelling follow the Pareto principle. Around 80% of the effort in the modelling process goes towards data cleaning and wrangling, while only 20% of the time and effort goes into implementing the model and getting the prediction. However, the meaty part of the modelling that is rich with almost 80% of results and insights is undoubtedly the implementation of the model. This information can be better represented as a matrix, which can be called a task matrix that will look something similar to the following figure:

![Task Matrix for Predictive Modelling](image)

Fig. 1.2: Task matrix: split of time spent on data cleaning and modelling and their final contribution to the model
Getting Started with Predictive Modelling

Many of the data cleaning and exploration chores can be automated because they are alike most of the times, irrespective of the data. The part that needs a lot of human thinking is the implementation of a model, which is what makes the bulk of this book.

Applications and examples of predictive modelling

In the introductory section, data has been compared with oil. While oil has been the primary source of energy for the last couple of centuries and the legends of OPEC, Petrodollars, and Gulf Wars have set the context for the oil as a begrudged resource; the might of data needs to be demonstrated here to set the premise for the comparison. Let us glance through some examples of predictive analytics to marvel at the might of data.

LinkedIn's "People also viewed" feature

If you are a frequent LinkedIn user, you might be familiar with LinkedIn's "People also viewed" feature.

What it does?

Let's say you have searched for some person who works at a particular organization and LinkedIn throws up a list of search results. You click on one of them and you land up on their profile. In the middle-right section of the screen, you will find a panel titled "People Also Viewed"; it is essentially a list of people who either work at the same organization as the person whose profile you are currently viewing or the people who have the same designation and belong to same industry.

Isn't it cool? You might have searched for these people separately if not for this feature. This feature increases the efficacy of your search results and saves your time.

How is it done?

Are you wondering how LinkedIn does it? The rough blueprint is as follows:

- LinkedIn leverages the search history data to do this. The model underneath this feature plunges into a treasure trove of search history data and looks at what people have searched next after finding the correct person they were searching for.
• This event of searching for a particular second person after searching for a particular first person has some probability. This will be calculated using all the data for such searches. The profiles with the highest probability of being searched (based on the historical data) are shown in the "People Also Viewed" section.

• This probability comes under the ambit of a broad set of rules called **Association Rules**. These are very widely used in Retail Analytics where we are interested to know what a group of products will sell together. In other words, what is the probability of buying a particular second product given that the consumer has already bought the first product?

**Correct targeting of online ads**

If you browse the Internet, which I am sure you must be doing frequently, you must have encountered online ads, both on the websites and smartphone apps. Just like the ads in the newspaper or TV, there is a publisher and an advertiser for online ads too. The publisher in this case is the website or the app where the ad will be shown while the advertiser is the company/organization that is posting that ad.

The ultimate goal of an online ad is to be clicked on. Each instance of an ad display is called an impression. The number of clicks per impression is called **Click Through Rate** and is the single most important metric that the advertisers are interested in. The problem statement is to determine the list of publishers where the advertiser should publish its ads so that the Click Through Rate is the maximum.

**How is it done?**

• The historical data in this case will consist of information about people who visited a certain website/app and whether they clicked the published ad or not. Some or a combination of classification models, such as Decision Trees, and Support Vector Machines are used in such cases to determine whether a visitor will click on the ad or not, given the visitor's profile information.

• One problem with standard classification algorithms in such cases is that the Click Through Rates are very small numbers, of the order of less than 1%. The resulting dataset that is used for classification has a very sparse positive outcome. The data needs to be downsampled to enrich the data with positive outcomes before modelling.

The logistical regression is one of the most standard classifiers for situations with binary outcomes. In banking, whether a person will default on his loan or not can be predicted using logistical regression given his credit history.
Santa Cruz predictive policing
Based on the historical data consisting of the area and time window of the occurrence of a crime, a model was developed to predict the place and time where the next crime might take place.

How is it done?
- A decision tree model was created using the historical data. The prediction of the model will foretell whether a crime will occur in an area on a given date and time in the future.
- The model is consistently recalibrated every day to include the crimes that happened during that day.

The good news is that the police are using such techniques to predict the crime scenes in advance so that they can prevent it from happening. The bad news is that certain terrorist organizations are using such techniques to target the locations that will cause the maximum damage with minimal efforts from their side. The good news again is that this strategic behavior of terrorists has been studied in detail and is being used to form counter-terrorist policies.

Determining the activity of a smartphone user using accelerometer data
The accelerometer in a smartphone measures the acceleration over a period of time as the user indulges in various activities. The acceleration is measured over the three axes, X, Y, and Z. This acceleration data can then be used to determine whether the user is sleeping, walking, running, jogging, and so on.

How is it done?
- The acceleration data is clustered based on the acceleration values in the three directions. The values of the similar activities cluster together.
- The clustering performs well in such cases if the columns contributing the maximum to the separation of activities are also included while calculating the distance matrix for clustering. Such columns can be found out using a technique called Singular Value Decomposition.
Sport and fantasy leagues

*Moneyball*, anyone? Yes, the movie. The movie where a statistician turns the fortunes of a poorly performing baseball team, Oak A, by developing an algorithm to select players who were cheap to buy but had a lot of latent potential to perform.

How was it done?

- Bill James, using historical data, concluded that the older metrics used to rate a player, such as stolen balls, runs batted in, and batting average were not very useful indicators of a player's performance in a given match. He rather relied on metrics like on-base percentage and sluggish percentage to be a better predictor of a player's performance.

- The chief statistician behind the algorithms, Bill James, compiled the data for performance of all the baseball league players and sorted them for these metrics. Surprisingly, the players who had high values for these statistics also came at cheaper prices.

This way, they gathered an unbeatable team that didn't have individual stars who came at hefty prices but as a team were an indomitable force. Since then, these algorithms and their variations have been used in a variety of real and fantasy leagues to select players. The variants of these algorithms are also being used by Venture Capitalists to optimize and automate their due diligence to select the prospective start-ups to fund.

Python and its packages – download and installation

There are various ways in which one can access and install Python and its packages. Here we will discuss a couple of them.

Anaconda

Anaconda is a popular Python distribution consisting of more than 195 popular Python packages. Installing Anaconda automatically installs many of the packages discussed in the preceding section, but they can be accessed only through an IDE called Spyder (more on this later in this chapter), which itself is installed on Anaconda installation. Anaconda also installs IPython Notebook and when you click on the IPython Notebook icon, it opens a browser tab and a Command Prompt.
Anaconda can be downloaded and installed from the following web address: http://continuum.io/downloads

Download the suitable installer and double click on the .exe file and it will install Anaconda. Two of the features that you must check after the installation are:

- IPython Notebook
- Spyder IDE

Search for them in the "Start" icon's search, if it doesn't appear in the list of programs and files by default. We will be using IPython Notebook extensively and the codes in this book will work the best when run in IPython Notebook.

IPython Notebook can be opened by clicking on the icon. Alternatively, you can use the Command Prompt to open IPython Notebook. Just navigate to the directory where you have installed Anaconda and then write `ipython notebook`, as shown in the following screenshot:

![Fig. 1.3: Opening IPython Notebook](image)

Standalone Python

You can download a Python version that is stable and is compatible to the OS on your system. The most stable version of Python is 2.7.0. So, installing this version is highly recommended. You can download it from https://www.python.org/ and install it.
Chapter 1

There are some Python packages that you need to install on your machine before you start predictive analytics and modelling. This section consists of a demo of installation of one such library and a brief description of all such libraries.

**Installing a Python package**

There are several ways to install a Python package. The easiest and the most effective is the one using pip. As you might be aware, pip is a package management system that is used to install and manage software packages written in Python. To be able to use it to install other packages, pip needs to be installed first.

**Installing pip**

The following steps demonstrate how to install pip. Follow closely!

1. Navigate to the webpage shown in the following screenshot. The URL address is https://pypi.python.org/pypi/pip:

   ![Downloading pip from the Python's official website](http://pypi.python.org/pypi/pip)

   Downloading pip from the Python's official website
2. Download the `pip-7.0.3.tar.gz` file and unzip in the folder where Python is installed. If you have Python v2.7.0 installed, this folder should be `C:\Python27`:

![Unzipping the .tar.gz file for pip in the correct folder](image)

3. On unzipping the previously mentioned file, a folder called `pip-7.0.3` is created. Opening that folder will take you to the screen similar to the one in the preceding screenshot.

4. Open the CMD on your computer and change the current directory to the current directory in the preceding screenshot that is `C:\Python27\pip-7.0.3` using the following command:

   ```
   cd C:\Python27\pip-7.0.3
   ```

5. The result of the preceding command is shown in the following screenshot:

![Navigating to the directory where pip is installed](image)

6. Now, the current directory is set to the directory where setup file for pip (`setup.py`) resides. Write the following command to install pip:

   ```
   python setup.py install
   ```
7. The result of the preceding command is shown in the following screenshot:

![Installing pip using a command line](image1)

Once pip is installed, it is very easy to install all the required Python packages to get started.

**Installing Python packages with pip**

The following are the steps to install Python packages using pip, which we just installed in the preceding section:

1. Change the current directory in the command prompt to the directory where the Python v2.7.0 is installed that is: C:\Python27.
2. Write the following command to install the package:
   ```bash
   pip install package-name
   ```
3. For example, to install pandas, you can proceed as follows:

   ![Installing a Python package using a command line and pip](image2)

4. Finally, to confirm that the package has installed successfully, write the following command:
   ```bash
   python -c "import pandas"
   ```
5. The result of the preceding command is shown in the following screenshot:

   ![Checking whether the package has installed correctly or not](image3)
If this doesn't throw up an error, then the package has been installed successfully.

**Python and its packages for predictive modelling**

In this section, we will discuss some commonly used packages for predictive modelling.

**pandas**: The most important and versatile package that is used widely in data science domains is pandas and it is no wonder that you can see `import pandas` at the beginning of any data science code snippet, in this book, and anywhere in general. Among other things, the `pandas` package facilitates:

- The reading of a dataset in a usable format (data frame in case of Python)
- Calculating basic statistics
- Running basic operations like sub-setting a dataset, merging/concatenating two datasets, handling missing data, and so on

The various methods in `pandas` will be explained in this book as and when we use them.

To get an overview, navigate to the official page of `pandas` here: [http://pandas.pydata.org/index.html](http://pandas.pydata.org/index.html)

**NumPy**: NumPy, in many ways, is a MATLAB equivalent in the Python environment. It has powerful methods to do mathematical calculations and simulations. The following are some of its features:

- A powerful and widely used a N-d array element
- An ensemble of powerful mathematical functions used in linear algebra, Fourier transforms, and random number generation
- A combination of random number generators and an N-d array elements is used to generate dummy datasets to demonstrate various procedures, a practice we will follow extensively, in this book

matplotlib: matplotlib is a Python library that easily generates high-quality 2-D plots. Again, it is very similar to MATLAB.

- It can be used to plot all kind of common plots, such as histograms, stacked and unstacked bar charts, scatterplots, heat diagrams, box plots, power spectra, error charts, and so on
- It can be used to edit and manipulate all the plot properties such as title, axes properties, color, scale, and so on

To get an overview, navigate to the official page of matplotlib at:
http://matplotlib.org

IPython: IPython provides an environment for interactive computing.

It provides a browser-based notebook that is an IDE-cum-development environment to support codes, rich media, inline plots, and model summary. These notebooks and their content can be saved and used later to demonstrate the result as it is or to save the codes separately and execute them. It has emerged as a powerful tool for web based tutorials as the code and the results flow smoothly one after the other in this environment. At many places in this book, we will be using this environment.

To get an overview, navigate to the official page of IPython here
http://ipython.org/

Scikit-learn: scikit-learn is the mainstay of any predictive modelling in Python. It is a robust collection of all the data science algorithms and methods to implement them. Some of the features of scikit-learn are as follows:

- It is built entirely on Python packages like pandas, NumPy, and matplotlib
- It is very simple and efficient to use
- It has methods to implement most of the predictive modelling techniques, such as linear regression, logistic regression, clustering, and Decision Trees
- It gives a very concise method to predict the outcome based on the model and measure the accuracy of the outcomes

To get an overview, navigate to the official page of scikit-learn here:

Python packages, other than these, if used in this book, will be situation based and can be installed using the method described earlier in this section.
IDEs for Python

The IDE or the Integrated Development Environment is a software that provides the source-code editor cum debugger for the purpose of writing code. Using these software, one can write, test, and debug a code snippet before adding the snippet in the production version of the code.

**IDLE**: IDLE is the default Integrated Development Environment for Python that comes with the default implementation of Python. It comes with the following features:

- Multi-window text-editor with auto-completion, smart-indent, syntax, and keyword highlighting
- Python shell with syntax highlighting

IDLE is widely popular as an IDE for beginners; it is simple to use and works well for simple tasks. Some of the issues with IDLE are bad output reporting, absence of line numbering options, and so on. As a result, advanced practitioners move on to better IDEs.

**IPython Notebook**: IPython Notebook is a powerful computational environment where code, execution, results, and media can co-exist in one single document. There are two components of this computing environment:

- **IPython Notebook**: Web applications containing code, executions, plots, and results are stored in different cells; they can be saved and edited as and when required
- **Notebook**: It is a plain text document meant to record and distribute the result of a computational analysis

The IPython documents are stored with an extension `.ipynb` in the directory where it is installed on the computer.

Some of the features of IPython Notebook are as follows:

- Inline figure rendering of the `matplotlib` plots that can be saved in multiple formats (JPEG, PNG).
- Standard Python syntax in the notebook can be saved as a Python script.
- The notebooks can be saved as HTML files and `.ipynb` files. These notebooks can be viewed in browsers and this has been developed as a popular tool for illustrated blogging in Python. A notebook in IPython looks as shown in the following screenshot:
Spyder: Spyder is a powerful scientific computing and development environment for Python. It has the following features:

- Advanced editing, auto-completion, debugging, and interactive testing
- Python kernel and code editor with line numbering in the same screen
- Preinstalled scientific packages like NumPy, pandas, scikit-learn, matplotlib, and so on.
In some ways, Spyder is very similar to RStudio environment where text editing and interactive testing go hand in hand:

The interface of Spyder IDE

In this book, IPython Notebook and Spyder have been used extensively. IDLE has been used from time to time and some people use other environments, such as Pycharm. Readers of this book are free to use such editors if they are more comfortable with them. However, they should make sure that all the required packages are working fine in those environments.
Summary
The following are some of the takeaways from this chapter:

• Social media and Internet of Things have resulted in an avalanche of data.
• Data is powerful but not in its raw form. The data needs to be processed and modelled.
• Organizations across the world and across the domains are using data to solve critical business problems. The knowledge of statistical algorithms, statistical tool, business context, and handling of historical data is vital to solve these problems using predictive modelling.
• Python is a robust tool to handle, process, and model data. It has an array of packages for predictive modelling and a suite of IDEs to choose from.

Let us enter the battlefield where Python is our weapon. We will start using it from the next chapter. In the next chapter, we will learn how to read data in various cases and do a basic processing.
Where to buy this book

You can buy Learning Predictive Analytics with Python from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.