UNIT-1

Introduction, Toolboxes: Python, fundamental libraries for data Scientists. Integrated development environment (IDE). Data operations: Reading, selecting, filtering, manipulating, sorting, grouping, rearranging, ranking, and plotting.

Introduction to Data Science

1.1 What is Data Science?

You have, no doubt, already experienced data science in several forms. When you are looking for information on the web by using a search engine or asking your mobile phone for directions, you are interacting with data science products. Data science has been behind resolving some of our most common daily tasks for several years.

Most of the scientific methods that power data science are not new and they havebeen out there, waiting for applications to be developed, for a long time. Statistics is an old science that stands on the shoulders of eighteenth-century giants such as Pierre Simon Laplace (1749–1827) and Thomas Bayes (1701–1761). Machine learning isyounger, but it has already moved beyond its infancy and can be considered a well- established discipline. Computer science changed our lives several decades ago andcontinues to do so; but it cannot be considered new.

So, why is data science seen as a novel trend within business reviews, in technologyblogs, and at academic conferences?

The novelty of data science is not rooted in the latest scientific knowledge, but in a disruptive change in our society that has been caused by the evolution of technology: datification. Datification is the process of rendering into data aspects of the world that have never been quantified before. At the personal level, the list of datified conceptsis very long and still growing: business networks, the lists of books we are reading,the films we enjoy, the food we eat, our physical activity, our purchases, our drivingbehavior, and so on. Even our thoughts are datified when we publish them on our favorite social network; and in a not so distant future, your gaze could be datified by wearable vision registering devices. At the business level, companies are datifying semi-structured data that were previously discarded: web activity logs, computer network activity, machinery signals, etc. Nonstructured data, such as written reports,e-mails, or voice recordings, are now being stored not only for archive purposes butalso to be analyzed.

However, datification is not the only ingredient of the data science revolution. The other ingredient is the democratization of data analysis. Large companies such as Google, Yahoo, IBM, or SAS were the only players in this field when data science had no name. At the beginning of the century, the huge computational resources of those companies allowed them to take advantage of datification by using analytical techniques to develop innovative products and even to take decisions about their own business. Today, the analytical gap between those companies and the rest of the world (companies and people) is shrinking. Access to cloud computing allows any individual to analyze huge amounts of data in short periods of time. Analytical knowledge is free and most of the crucial algorithms that are needed to create a solution can be found, because open-source development is the norm in this field. As a result, the possibility of using rich data to take evidence-based decisions is open to virtually any person or company.

Data science is commonly defined as a methodology by which actionable insights can be inferred from data. This is a subtle but important difference with respect to previous approaches to data analysis, such as business intelligence or exploratory statistics. Performing data science is a task with an ambitious objective: the produc-tion of beliefs informed by data and to be used as the basis of decision-making. In the absence of data, beliefs are uninformed and decisions, in the best of cases, are based on best practices or intuition. The representation of complex environments by rich data opens up the possibility of applying all the scientific knowledge we have regarding how to infer knowledge from data.

In general, data science allows us to adopt four different strategies to explore theworld using data:

1. *Probing reality*. Data can be gathered by passive or by active methods. In the latter case, data represents the response of the world to our actions. Analysis of those responses can be extremely valuable when it comes to taking decisions about our subsequent actions. One of the best examples of this strategy is the use of A/B testing for web development: What is the best button size and color? The best answer can only be found by probing the world.
2. *Pattern discovery*. Divide and conquer is an old heuristic used to solve complex problems; but it is not always easy to decide how to apply this common sense to problems. Datified problems can be analyzed automatically to discover useful patterns and natural clusters that can greatly simplify their solutions. The use of this technique to profile users is a critical ingredient today in such importantfields as programmatic advertising or digital marketing.
3. *Predicting future events*. Since the early days of statistics, one of the most important scientific questions has been how to build robust data models that are capa- ble of predicting future data samples. Predictive analytics allows decisions to be taken in response to future events, not only reactively. Of course, it is not possible to predict the future in any environment and there will always be unpre- dictable events; but the identification of predictable events represents valuableknowledge. For example, predictive analytics can be used

to optimize the tasks

planned for retail store staff during the following week, by analyzing data suchas weather, historic sales, traffic conditions, etc.

4. *Understanding people and the world*. This is an objective that at the moment is beyond the scope of most companies and people, but large companies and governments are investing considerable amounts of money in research areas such as understanding natural language, computer vision, psychology and neu- roscience. Scientific understanding of these areas is important for data science because in the end, in order to take optimal decisions, it is necessary to know the real processes that drive people's decisions and behavior. The development of deep learning methods for natural language understanding and for visual object recognition is a good example of this kind of research.

Toolboxes for Data Scientists

Introduction

In this chapter, first we introduce some of the tools that data scientists use. The toolbox of any data scientist, as for any kind of programmer, is an essential ingredient for success and enhanced performance. Choosing the right tools can save a lot of time and thereby allow us to focus on data analysis.

The most basic tool to decide on is which programming language we will use. Many people use only one programming language in their entire life: the first and only one they learn. For many, learning a new language is an enormous task that, if at all possible, should be undertaken only once. The problem is that some languages are intended for developing high-performance or production code, such as C, C++, or Java, while others are more focused on prototyping code, among these the best known are the so-called scripting languages: Ruby, Perl, and Python. So, depending on the first language you learned, certain tasks will, at the very least, be rather tedious. The main problem of being stuck with a single language is that many basic tools simply will not be available in it, and eventually you will have either to reimplementthem or to create a bridge to use some other language just for a specific task.

In conclusion, you either have to be ready to change to the best language for each task and then glue the results together, or choose a very flexible language with a rich ecosystem (e.g., third-party open-source libraries). In this book we have selected Python as the programming language.

Why Python?

Python[1] is a mature programming language but it also has excellent properties for newbie programmers, making it ideal for people who have never programmed before. Some of the most remarkable of those properties are easy to read code, suppression of non-mandatory delimiters, dynamic typing, and dynamic memory usage. Pythonis an interpreted language, so the code is executed immediately in the Python con- sole without needing the compilation step to machine language. Besides the Python console (which comes included with any Python installation) you can find other in-teractive consoles, such as IPython,[2] which give you a richer environment in whichto execute your Python code.

Currently, Python is one of the most flexible programming languages. One of its main characteristics that makes it so flexible is that it can be seen as a multiparadigm language. This is especially useful for people who already know how to program with other languages, as they can rapidly start programming with Python in the same way. For example, Java programmers will feel comfortable using Python as it supports the object-oriented paradigm, or C programmers could mix Python and C code using *cython*. Furthermore, for anyone who is used to programming in functional languages such as Haskell or Lisp, Python also has basic statements for functional programming in its own core library.

In this book, we have decided to use Python language because, as explained before, it is a mature language programming, easy for the newbies, and can be used as a specific platform for data scientists, thanks to its large ecosystem of scientific libraries and its high and vibrant community. Other popular alternatives to Python for data scientists are R and MATLAB/Octave.

Fundamental Python Libraries for Data Scientists

The Python community is one of the most active programming communities with a huge number of developed toolboxes. The most popular Python toolboxes for any data scientist are NumPy, SciPy, Pandas, and Scikit-Learn.

Numeric and Scientific Computation: NumPy and SciPy

NumPy[3] is the cornerstone toolbox for scientific computing with Python. NumPy provides, among other things, support for multidimensional arrays with basic oper-ations on them and useful linear algebra functions. Many toolboxes use the NumPyarray representations as an efficient basic data structure. Meanwhile, *SciPy* provides a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics, and much more. Another core toolbox in SciPyis the plotting library *Matplotlib*. This toolbox has many tools for data visualization.

SCIKIT-Learn: Machine Learning in Python

Scikit-learn[4] is a machine learning library built from NumPy, SciPy, and Matplotlib. Scikit-learn offers simple and efficient tools for common tasks in data analysis such as classification, regression, clustering, dimensionality reduction, model selection, and preprocessing.

PANDAS: Python Data Analysis Library

Pandas[5] provides high-performance data structures and data analysis tools. The keyfeature of Pandas is a fast and efficient DataFrame object for data manipulation withintegrated indexing. The DataFrame structure can be seen as a spreadsheet which offers very flexible ways of working with it. You can easily transform any dataset in the way you want, by reshaping it and adding or removing columns or rows. It also provides high-performance functions for aggregating, merging, and joining dataset-
s. Pandas also has tools for importing and exporting data from different formats: comma-separated value (CSV), text files, Microsoft Excel, SQL databases, and the fast HDF5 format. In many situations, the data you have in such formats will not be complete or totally structured. For such cases, Pandas offers handling of missing data and intelligent data alignment. Furthermore, Pandas provides a convenientMatplotlib interface.

Data Science Ecosystem Installation

Before we can get started on solving our own data-oriented problems, we will need toset up our programming environment. The first question we need to answer concerns

Python language itself. There are currently two different versions of Python: Python 2.X and Python 3.X. The differences between the versions are important, so there isno compatibility between the codes, i.e., code written in Python 2.X does not workin Python 3.X and vice versa. Python 3.X was introduced in late 2008; by then, a lotof code and many toolboxes were already deployed using Python 2.X (Python 2.0 was initially introduced in 2000). Therefore, much of the scientific community didnot change to Python 3.0 immediately and they were stuck with Python 2.7. By now, almost all libraries have been ported to Python 3.0; but Python 2.7 is still maintained, so one or another version can be chosen. However, those who already have a large amount of code in 2.X rarely change to Python 3.X. In our examples throughout thisbook we will use Python 2.7.

Once we have chosen one of the Python versions, the next thing to decide is