1
The Predictive Analytics Process

This will be the only conceptual chapter of the book; you may want to start coding and building predictive models from the start, but trust me, we need a common understanding of the fundamental concepts that we will use in the rest of the book. First, we will discuss in detail what predictive analytics is, then we will define some of the most important concepts of this exciting field. With those concepts as a foundation, we will go on to provide a quick overview of the stages in the predictive analytics process, and finally briefly talk about them, as we will devote entire chapters to each of them in the rest of the book.

The following topics will be covered in this chapter:

- What is predictive analytics?
- A review of important concepts of predictive analytics
- The predictive analytics process
- A quick tour of Python's data science stack

Technical requirements

Although this is mostly a conceptual chapter, you need at least the following software to follow the code snippets:

- Python 3.6 or higher
- Jupyter Notebook
- Recent versions of the following Python libraries: NumPy and matplotlib

I strongly recommend that you install Anaconda Distribution (go to `https://www.anaconda.com/`) so you have most of the software we will use in the rest of the book. If you are not familiar with Anaconda, we will talk about it later in the chapter, so please keep reading.

What is predictive analytics?

With the exponentially growing amounts of data the world has been observing, especially in the last decade, the number of related technologies and terms also started growing at a faster rate. Suddenly, people in industry, media, and academia started talking (sometimes maybe too much) about big data, data mining, analytics, machine learning, data science, data engineering, statistical learning, artificial intelligence, and many other related terms, and of course one of those terms is **predictive analytics**, the subject of this book.

There is still a lot of confusion about these terms and exactly what they mean, because they are relatively new. As there is some overlap between them, for our purposes, instead of attempting to define all these terms, I will give a working definition that we can keep in mind as we work through the content of this book. You can also use this definition to find out what predictive analytics is:

> *Predictive analytics is an applied field that uses a variety of quantitative methods that make use of data in order to make predictions*

Let's break down and analyze this definition:

- **Is an applied field**: There is no such thing as *Theoretical Predictive Analytics*; the field of predictive analytics is always used to solve problems and it is being applied in virtually every industry and domain: finance, telecommunications, advertising, insurance, healthcare, education, entertainment, banking, and so on. So keep in mind that you will be always using predictive analytics to solve problems within a particular domain, which is why having the context of the problem and *domain knowledge* is a key aspect of doing predictive analytics. We will discuss more about this in the next chapter.

- **Uses a variety of quantitative methods**: When doing predictive analytics, you will be a user of the techniques, theorems, best practices, empirical findings, and theoretical results of mathematical sciences such as computer science and statistics and other sub-fields of those disciplines, and of mathematics such as optimization, probability theory, linear algebra, artificial intelligence, machine learning, deep learning, algorithms, data structures, statistical inference, visualization, and Bayesian inference, among others. I would like to stress that you will be a user of these many sub-fields; they will give you the analytical tools you will use to solve problems and you won't be producing any theoretical results when doing predictive analytics, but your results and conclusions must be consistent with the established theoretical results. This means that you must be able to use the tools properly, and for that, you need the proper conceptual foundation: you need to feel comfortable with the basics of some of the mentioned fields to be able to do predictive analytics correctly and rigorously. In the following chapter, we will discuss many of these fundamental topics at a high and intuitive level and we will provide you with proper sources if you need to go deeper in any of these topics.
- **That makes use of data**: If quantitative methods are the tools of predictive analytics, then data is the raw material out of which you will (literally) build the models. A key aspect of predictive analytics is the use of data to extract useful information from it. Using data has been proven highly valuable for guiding decision-making: all over the world, organizations of all types are adopting a data-driven approach for making decisions at all levels; rather than relying on intuition or *gut feeling*, organizations rely increasingly on data. Predictive analytics is another application that uses data, in this case, to make predictions that can then be used to solve problems which can have a measurable impact.

Since the operations and manipulations that need to be done in predictive analytics (or any other type of advanced analytics) usually go well beyond what a spreadsheet allows us to do, to properly carry out predictive analytics we need a programming language. Python and R have become popular choices (although people do use different ones, such as Julia, for instance).

In addition, you may need to work directly with the data storage systems such as relational or non-relational databases or any of the big data storage solutions, which is why you may need to be familiar with things such as SQL and Hadoop; however, since what is done with those technologies is out of the scope for this book, we won't discuss them any further. We will start all the examples in the book assuming that we are given the data from a storage system and we won't be concerned with how the data was extracted. Starting from raw data, we will see some of the manipulations and transformations that are commonly done within the predictive analytics process. We will do everything using Python and related tools and we'll delve deeper into these manipulations in the coming sections and chapters.

- **To make predictions**: The last part of the definition seems straightforward, however, one clarification is needed here—in the context of predictive analytics, a *prediction* is an unknown event, not necessarily about the future as is understood in the colloquial sense. For instance, we can build a predictive model that is able to "predict", if a patient has the disease X using his clinical data. Now, when we gather the patient's data, the disease X *is already present or not*, so we are not "predicting" if the patient *will have the disease X in the future*; the model is giving an assessment (an educated guess) about the unknown event "the patient has disease X". Sometimes, of course, the prediction will actually be about the future, but keep in mind that won't be necessarily the case.

Let's take a look at some of the most important concepts in the field; we need a firm grasp of them before moving forward.

Reviewing important concepts of predictive analytics

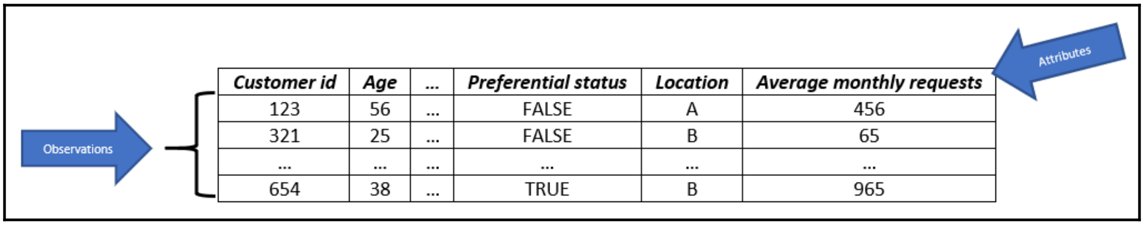
In this section, we introduce and clarify the meaning of some of the terms we will be using throughout the book. Part of what is confusing for beginners in this field is sometimes the terminologies. There are many words for the same concept. One extreme example is *variable*, *feature*, *attribute*, *independent variable*, *predictor*, *regressor*, *covariate*, *explanatory variable*, *input*, and *factor*: they all may refer to the same thing! The reason for this (I must admit) shameful situation is that many practitioners of predictive analytics come from different fields (statistics, econometrics, computer science, operations research, and so on) and their community has its own way to name things, so when they come to predictive analytics they bring their vocabulary with them. But don't worry, you'll get used to it.

OK, now let's look at some of the fundamental concepts. Keep in mind that the terms won't be defined too formally, and you don't need to memorize them word by word (nobody will test you!). My intention is for us to have a common understanding of what we will be talking about. Since we have seen that data is the raw material of predictive analytics, let's define some key concepts:

- **Data**: Any record that is captured and stored and that is meaningful in some context.
- **Unit of observation**: The entity that is the subject of analysis. Although many a time it will be clear from the context, sometimes it can be tricky to define (especially when talking at a high level with non-technical people). Suppose that you are asked to analyze "sales data" for a set of stores in a supermarket chain. There can be many units of observation that can be defined for this (vaguely defined) task: stores, cash registers, transactions, days, and so on. Once you know what the unit of observation is (customers, houses, patients, cities, cells, rocks, stars, books, products, transactions, tweets, websites, and so on) you can start asking about their attributes.
- **Attribute**: A characteristic of a unit of analysis. If our unit of analysis is a patient, then examples of attributes of the patient could be age, height, weight, body mass index, cholesterol level, and so on.
- **Data point, sample, observation, and instance**: A single unit of observation with all its available attributes.
- **Dataset**: A collection of data points, usually in a table format; think of a relational database table or a spreadsheet.

For many problems, the data comes in an unstructured format, such as video, audio, a set of tweets, and blog posts. However, in predictive analytics, when we talk about a dataset, we often implicitly mean a structured dataset: a table or a set of mutually related tables. It is very likely that a big portion of your time at your job when doing predictive analytics is spent transforming unstructured raw data into a structured dataset.

From here, when we refer to a dataset, we will be talking about a single table; although in the real world a dataset may consist of multiple tables, when we do predictive modeling we do it with a single table. The typical table looks like this:


Observations
Attributes

Customer id	*Age*	*...*	*Preferential status*	*Location*	*Average monthly requests*
123	56	...	FALSE	A	456
321	25	...	FALSE	B	65
...	...	...	...	...	...
654	38	...	TRUE	B	965

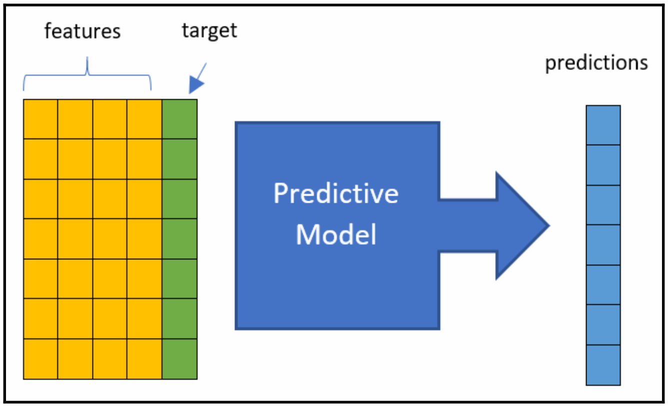
In the former dataset, our unit of observation is a *customer*, the entity of interest. Every row is an observation or a data point and, as you can see, each data point has a number of attributes (**Customer ID**, **Age**, **Preferential status**, and so on). Now, let's talk about the vocabulary used for modeling in relation to a dataset: first, every column in our dataset is considered a *variable* in the mathematical sense: their values are subject to change; they can *vary* from one data point to another data point. One of the most important things to know about the variables in a dataset is their types, which can be the following:

- **Categorical variables**: Variables that can be accepted as values with only a finite number of categories such as gender, country, type of transaction, age group, marital status, movie genre, and so on. Within this type of variables there are two sub-types:
    - **Ordinal variables**: When the categories have some natural ordering: for instance, age groups (21–30, 31–40, 41–50, 51+) or shirt size (small, medium, large)
    - **Nominal variables**: Those categorical variables whose values have no meaningful order
- **Numerical variables**: Variables whose values can vary in some defined interval. There are two sub-types, although the distinction in most cases won't be as important:
    - **Continuous variables**: Those that in principle can take any value within an interval: the height of a person, stock prices, the mass of a star, and credit card balance are examples of continuous variables
    - **Integer variables**: Those that can take only values that are integer numbers: number of children, age (if measured in years), the number of rooms in a house, and so on

One of the columns in our dataset plays a very important role: the one that we are interested in predicting. We call this column *target*, *dependent variable*, *response*, *outcome*, and *output variable*: the quantity or event that is being predicted. It is usually denoted by **y** and it is one of the columns in the dataset. We will use the term **target** throughout the book.

Once the target is identified, the rest of the columns are candidates to become *features*, *attributes*, *independent variables*, *predictors*, *regressors*, *explanatory variables*, and *inputs*: the columns in our dataset that will be used to predict the target. We will use the terms **variables** and **feature** throughout the book.

Finally, we can give a definition of **Predictive Model**: a method that uses the features to predict the target. It can also be thought of like a mathematical function: a predictive model takes inputs, meaning the set of features, the target, and outputs the predictions for the values of the target. At a high level, one way to think about a predictive model is like this:



This diagram is limited (and some might say it is even wrong), but for now I think it will give you a general idea of what a predictive model is. We will, of course, delve deeper into the details of predictive models and we will build many of them in the following chapters.

Now that we have a clear understanding of what predictive analytics is, and some of the most important terminology we will be using in the book, it is time to take a look at how it is done in practice: the predictive analytics process.

The predictive analytics process

There is a common misunderstanding about predictive analytics: that it is all about models. In fact, that is actually just part of doing predictive analytics. Practitioners of the field have established certain standard phrases that different authors refer to by different names. However, the order of the stages is logical and the relationships between them are well understood. In fact, this book has been organized in the logical order of these stages. Here they are:

1. Problem understanding and definition
2. Data collection and preparation
3. Data understanding using **exploratory data analysis** (**EDA**)
4. Model building
5. Model evaluation
6. Communication and/or deployment

We will dig deeper into all of them in the following chapters. For now, let's provide a brief overview of what every stage is about. I like to think about each of these phases as having a defined goal.

Problem understanding and definition

Goal: Understand the problem and how the potential solution would look. Also, define the requirements for solving the problem.

This is the first stage in the process. This is a key stage because here we establish together with the stakeholders what the objectives of the predictive model are—which is the problem that needs to be solved and how the *solution* looks from the business perspective.

In this phase, you also establish explicitly the requirements for the project. The requirements should be in terms of *inputs*: what the data needed for producing the solution is, in what format it is needed, how much data is needed, and so on. You also discuss what the outputs of the analysis and predictive model will look like and how they provide solutions for the problems that are being discussed. We will discuss much more about this phase in the next chapter.

Data collection and preparation

Goal: Get a dataset that is ready for analysis.

This phase is where we take a look at the data that is available. Depending on the project, you will need to interact with the database administrators and ask them to provide you with the data. You may also need to rely on many different sources to get the data that is needed. Sometimes, the data may not exist yet and you may be part of the team that comes up with a plan to collect it. Remember, the goal of this phase is to have a dataset you will be using for building the predictive model.

In the process of getting the dataset, potential problems with the data may be identified, which is why this phase is, of course, very closely related with the previous one. While performing the tasks for getting the dataset ready, you will go back and forth between this and the former phase as you may realize that the available data is not enough to solve the proposed problem as was formulated in the business understanding phase, so you may need to go back to the stakeholders and discuss the situation and maybe reformulate the problem and solution.

While building the dataset, you may notice some problems with some of the features. Maybe one column has a lot of missing values or the values have not been properly encoded. Although in principle it would be great to deal with problems such as missing values and outliers in this phase, that is often not the case, which is why there isn't a hard boundary between this phase and the next phase: EDA.

Dataset understanding using EDA

Goal: Understand your dataset.

Once you have collected the dataset, it is time for you to start understanding it using **EDA** which is a combination of numerical and visualization techniques that allow us to understand different characteristics of our dataset, its variables, and the potential relationship between them. The limits between this phase and the previous and next ones are often blurry, so you may think that your dataset is ready for analysis, but when you start your analysis you may realize that you have got five months of historical data from one source and two months from another source, or, for instance, you may find that three features are redundant or that you may need to combine some features to create a new one. So, after a few trips back to the previews phase you may finally get your dataset ready for analysis.

Now it is time for you to start understanding your dataset by starting to answer questions like the following:

- What types of variables are there in the dataset?
- What do their distributions look like?
- Do we still have missing values?
- Are there redundant variables?
- What are the relationships between the features?
- Do we observe outliers?
- How do the different pairs of features correlate with each other?
- Do these correlations make sense?
- What is the relationship between the features and the target?

All the questions that you try to answer in this phase must be guided by the goal of the project: always keep in mind the problem you are trying to solve. Once you have a good understanding of the data, you will be ready for the next phase: model building.

Model building

Goal: Produce some predictive models that solve the problem.

Here is where you build many predictive models that you will then evaluate to pick the best one. You must choose the type of model that will be *trained* or *estimated.* The term *model training* is associated with machine learning and the term *estimation* is associated with statistics. The approach, type of model, and training/estimation process you will use must be absolutely determined by the problem you are trying to solve and the solution you are looking for.

How to build models with Python and its data science ecosystem is the subject of the majority of this book. We will take a look at different approaches: machine learning, deep learning, Bayesian statistics. After trying different approaches, types of models, and fine-tuning techniques, at the end of this phase you may end up with some models considered to be *finalists*, and from the most promising ones of which the candidate winner will emerge: the one that will produce the best solution.

Model evaluation

Goal: Choose the best model among a subset of the most promising ones and determine how good the model is in providing the solution.

Here is where you evaluate the subset of "finalists" to see how well they perform. Like every other stage in the process, the evaluation is determined by the problem to be solved. Usually, one or more main metrics will be used to evaluate how good the model performs. Depending on the project, other criteria may be considered when evaluating the model besides metrics, such as computational considerations, interpretability, user-friendliness, and methodology, among others. We will talk in depth about standard metrics and other considerations in `Chapter 7`, *Model Evaluation*. As with all the other stages, the criteria and metrics for model evaluation should be chosen considering the problem to be solved.

Please remember that the best model is not the fanciest, the most complex, the most mathematically impressive, the most computationally efficient, or the latest in the research literature: the best model is the one that solves the problem in the best possible way. So, any of the characteristics that we just talked about (fanciness, complexity, and so on) should not be considered when evaluating the model.

Communication and/or deployment

Goal: Use the predictive model and its results.

Finally, the model has been built, tested, and well evaluated: you did it! In the ideal situation, it solves the problem and its performance is great; now it is time to use it. How the model will be used depends on the project; sometimes the results and predictions will be the subject of a report and/or a presentation that will be delivered to key stakeholders, which is what we mean by communication—and, of course, good communication skills are very useful for this purpose.

Sometimes, the model will be incorporated as part of a software application: either web, desktop, mobile, or any other type of technology. In this case, you may need to interact closely with or even be part of the software development team that incorporates the model into the application. There is another possibility: the model itself may become a "data product". For example, a credit scoring application that uses customer data to calculate the chance of the customer defaulting on their credit card. We will produce one example of such data products in `Chapter 9`, *Implementing a Model with Dash*.

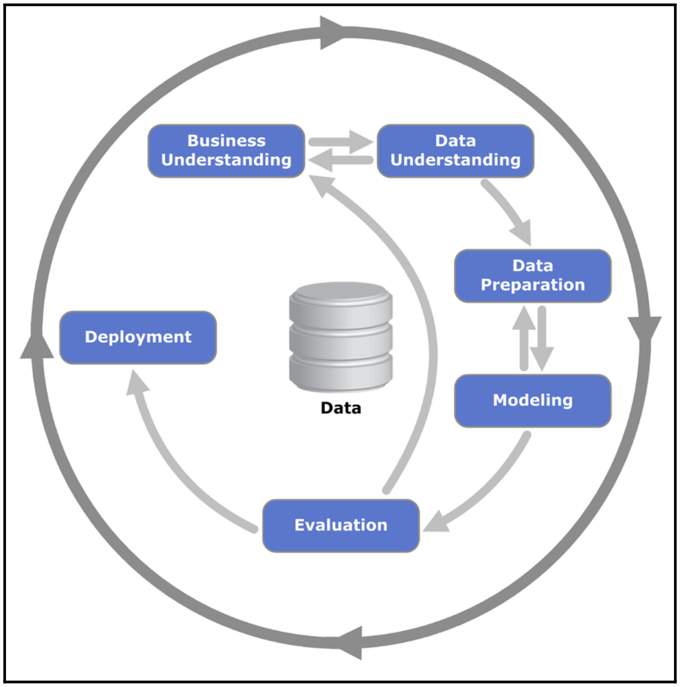
Although we have enumerated the stages in order, keep in mind that this is a highly iterative, non-linear process and you will be going back and forth between these stages; the frontiers between adjacent phases are blurry and there is always some overlap between them, so it is not important to place every task under some phase. For instance, when dealing with outliers, is it part of the *Data collection and preparation* phase or of the *Dataset understanding* phase? In practice, it doesn't matter, you can place it where you want; what matters is that it needs to be done!

Still, knowing the logical sequence of the stages is very useful when doing predictive analytics, as it helps with preparing and organizing the work, and it helps in setting the expectations for the duration of a project. The sequence of stages is logical in the sense that a previous stage is a prerequisite for the next: for example, you can't do model evaluation without having built a model, and after evaluation you may conclude that the model is not working properly so you go back to the *Model building* phase and come up with another one.

CRISP-DM and other approaches

Another popular framework for doing predictive analytics is the cross-industry standard process for data mining, most commonly known by its acronym, CRISP-DM, which is very similar to what we just described. This methodology is described in Wirth, R. & Hipp, J. (2000). In this methodology, the process is broken into six major phases, shown in the following diagram. The authors clarify that the sequence of the phases is not strict; although the arrows indicate the most frequent relationships between phases, those depend on the particularities of the project or the problem being solved. These are the phases of a predictive analytics project in this methodology:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment



By Kenneth Jensen-Own work based on `ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf` (Figure 1), CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=24930610`.

There are other ways to look at this process; for example, R. Peng (2016) describes the process using the concept of *Epicycles of Data Analysis*. For him, the epicycles are the following:

1. Develop expectations
2. Collect data
3. Match expectations with the data
4. State a question
5. Exploratory data analysis
6. Model building
7. Interpretation
8. Communication