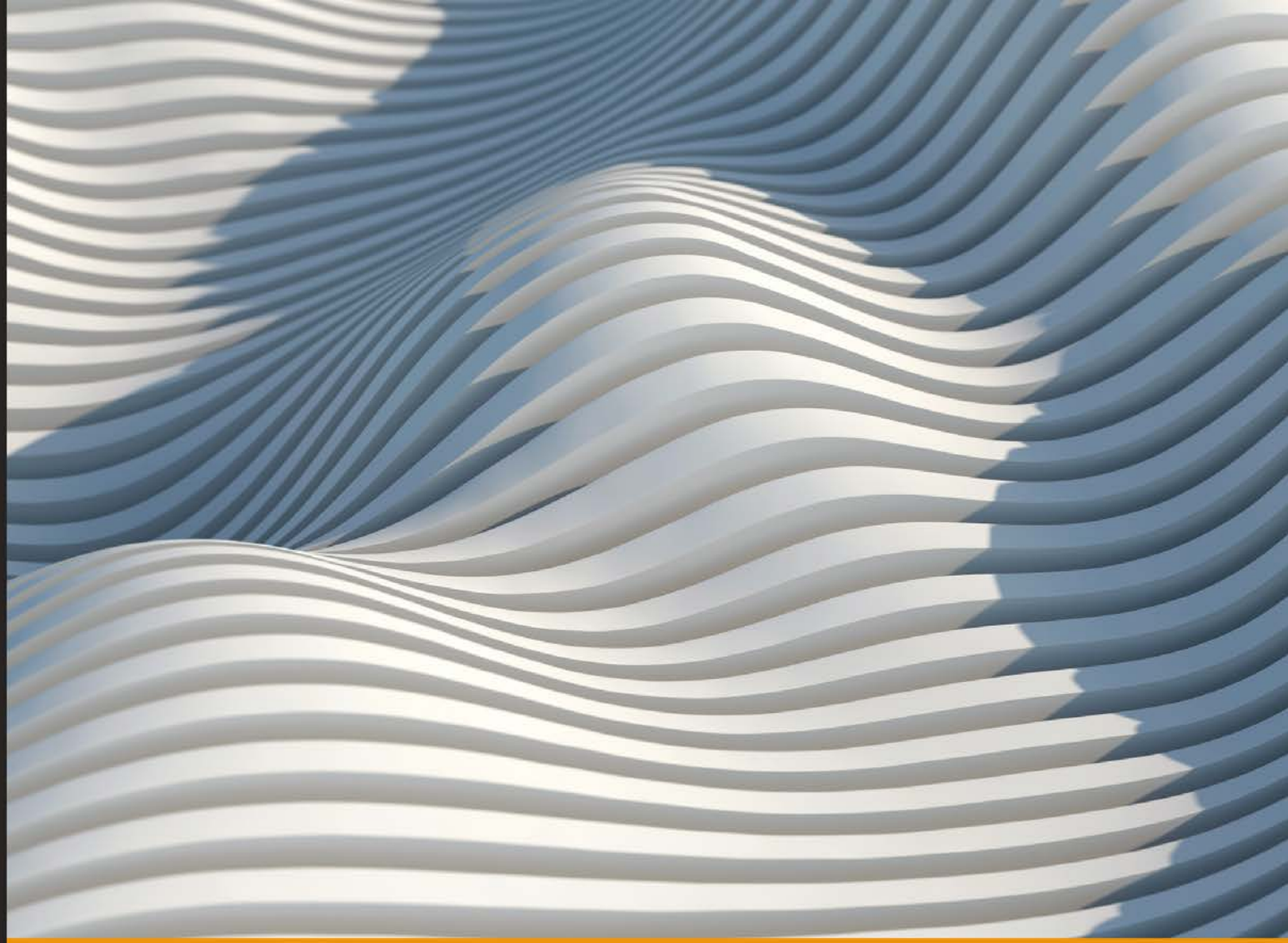


Community Experience Distilled

Mastering Machine Learning with R

Master machine learning techniques with R to deliver insights for complex projects

Cory Lesmeister

[PACKT] PUBLISHING open source* community experience distilled

Mastering Machine Learning with R

Master machine learning techniques with R to deliver insights for complex projects

Cory Lesmeister

[PACKT] PUBLISHING open source community experience distilled

BIRMINGHAM - MUMBAI

Mastering Machine Learning with R

First published: October 2015

Production reference: 1231015

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78398-452-7

www.packtpub.com

Credits

Author
Cory Lesmeister

Reviewers
Vikram Dhillon
Miro Kopecky
Pavan Narayanan
Doug Ortiz
Shivani Rao, PhD

Commissioning Editor
Kartikey Pandey

Acquisition Editor
Nadeem N. Bagban

Content Development Editor
Siddhesh Salvi

Technical Editor
Suwarna Rajput

Copy Editor
Tasneem Fatehi

Project Coordinator
Nidhi Joshi

Proofreader
Safis Editing

Indexer
Mariammal Chettiyar

Graphics
Disha Haria

Production Coordinator
Nilesh Mohite

Cover Work
Nilesh Mohite

About the Author

Cory Lesmeister currently works as an advanced analytics consultant for Clarity Solution Group, where he applies the methods in this book to solve complex problems and provide actionable insights. Cory spent 16 years at Eli Lilly and Company in sales, market research, Lean Six Sigma, marketing analytics, and new product forecasting. A former U.S. Army Reservist, Cory was in Baghdad, Iraq, in 2009 as a strategic advisor to the 29,000-person Iraqi oil police, where he supplied equipment to help the country secure and protect its oil infrastructure. An aviation aficionado, Cory has a BBA in aviation administration from the University of North Dakota and a commercial helicopter license. Cory lives in Carmel, IN, with his wife and their two teenage daughters.

About the Reviewers

Vikram Dhillon is a software developer, bioinformatics researcher, and software coach at the Blackstone LaunchPad in the University of Central Florida. He has been working on his own start-up involving healthcare data security. He lives in Orlando and regularly attends developer meetups and hackathons. He enjoys spending his spare time reading about new technologies such as the blockchain and developing tutorials for machine learning in game design. He has been involved in open source projects for over 5 years and writes about technology and start-ups at `opsbug.com`.

Miro Kopecky is a passionate JVM enthusiast from the first moment he joined Sun Microsystems in 2002. Miro truly believes in a distributed system design, concurrency, and parallel computing, which means pushing the system's performance to its limits without losing reliability and stability. He has been working on research of new data mining techniques in neurological signal analysis during his PhD studies. Miro's hobbies include autonomic system development and robotics.

I would like to thank my family and my girlfriend, Tanja, for their support during the reviewing of this book.

Pavan Narayanan is an applied mathematician and is experienced in mathematical programming, analytics, and web development. He has published and presented papers in algorithmic research to the Transportation Research Board, Washington DC and SUNY Research Conference, Albany, NY. An avid blogger at `https://datasciencehacks.wordpress.com`, his interests are exploring problem solving techniques—from industrial mathematics to machine learning. Pavan can be contacted at `pavan.narayanan@gmail.com`.

He has worked on books such as *Apache mahout essentials*, *Learning apache mahout*, and *Real-time applications development with Storm and Petrel*.

I would like to thank my family and God Almighty for giving me strength and endurance and the folks at Packt Publishing for the opportunity to work on this book.

Doug Ortiz is an independent consultant who has been architecting, developing, and integrating enterprise solutions throughout his whole career. Organizations that leverage his skillset have been able to rediscover and reuse their underutilized data via existing and emerging technologies such as Microsoft BI Stack, Hadoop, NOSQL Databases, SharePoint, Hadoop, and related toolsets and technologies.

Doug has experience in integrating multiple platforms and products. He has helped organizations gain a deeper understanding and value of their current investments in data and existing resources turning them into useful sources of information. He has improved, salvaged, and architected projects by utilizing unique and innovative techniques.

His hobbies include yoga and scuba diving. He is the founder of Illustris, LLC, and can be contacted at `dougortiz@illustris.org`.

Shivani Rao, PhD, is a machine learning engineer based in San Francisco and Bay Area working in areas of search, analytics, and machine learning. Her background and areas of interest are in the field of computer vision, image processing, applied machine learning, data mining, and information retrieval. She has also accrued industry experience in companies such as Nvidia , Google, and Box. Shivani holds a PhD from the Computer Engineering Department of Purdue University spanning areas of machine learning, information retrieval, and software engineering. Prior to that, she obtained a masters from the Computer Science and Engineering Department of the Indian Institute of Technology (IIT), Madras, majoring in Computer Vision and Image Processing.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit `www.PacktPub.com`.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.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 Packt books and eBooks.

`https://www2.packtpub.com/books/subscription/packtlib`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

"He who defends everything, defends nothing."

— Frederick the Great

Machine learning is a very broad topic. The following quote sums it up nicely: *The first problem facing you is the bewildering variety of learning algorithms available. Which one to use? There are literally thousands available, and hundreds more are published each year.* (Domingo, P., 2012.) It would therefore be irresponsible to try and cover everything in the chapters that follow because, to paraphrase Frederick the Great, we would achieve nothing.

With this constraint in mind, I hope to provide a solid foundation of algorithms and business considerations that will allow the reader to walk away and, first of all, take on any machine learning tasks with complete confidence, and secondly, be able to help themselves in figuring out other algorithms and topics. Essentially, if this book significantly helps you to help yourself, then I would consider this a victory. Don't think of this book as a destination but rather, as a path to self-discovery.

The world of R can be as bewildering as the world of machine learning! There is seemingly an endless number of R packages with a plethora of blogs, websites, discussions, and papers of various quality and complexity from the community that supports R. This is a great reservoir of information and probably R's greatest strength, but I've always believed that an entity's greatest strength can also be its greatest weakness. R's vast community of knowledge can quickly overwhelm and/or sidetrack you and your efforts. Show me a problem and give me ten different R programmers and I'll show you ten different ways the code is written to solve the problem. As I've written each chapter, I've endeavored to capture the critical elements that can assist you in using R to understand, prepare, and model the data. I am no R programming expert by any stretch of the imagination, but again, I like to think that I can provide a solid foundation herein.

Another thing that lit a fire under me to write this book was an incident that happened in the hallways of a former employer a couple of years ago. My team had an IT contractor to support the management of our databases. As we were walking and chatting about big data and the like, he mentioned that he had bought a book about machine learning with R and another about machine learning with Python. He stated that he could do all the programming, but all of the statistics made absolutely no sense to him. I have always kept this conversation at the back of my mind throughout the writing process. It has been a very challenging task to balance the technical and theoretical with the practical. One could, and probably someone has, turned the theory of each chapter to its own book. I used a heuristic of sorts to aid me in deciding whether a formula or technical aspect was in the scope, which was would this help me or the readers in the discussions with team members and business leaders? If I felt it might help, I would strive to provide the necessary details.

I also made a conscious effort to keep the datasets used in the practical exercises large enough to be interesting but small enough to allow you to gain insight without becoming overwhelmed. This book is not about big data, but make no mistake about it, the methods and concepts that we will discuss can be scaled to big data.

In short, this book will appeal to a broad group of individuals, from IT experts seeking to understand and interpret machine learning algorithms to statistical gurus desiring to incorporate the power of R into their analysis. However, even those that are well-versed in both IT and statistics—experts if you will—should be able to pick up quite a few tips and tricks to assist them in their efforts.

Machine learning defined

Machine learning is everywhere! It is used in web search, spam filters, recommendation engines, medical diagnostics, ad placement, fraud detection, credit scoring, and I fear in these autonomous cars that I hear so much about. The roads are dangerous enough now; the idea of cars with artificial intelligence, requiring *CTRL* + *ALT* + *DEL* every 100 miles, aimlessly roaming the highways and byways is just too terrifying to contemplate. But, I digress.

It is always important to properly define what one is talking about and machine learning is no different. The website, `machinelearningmastery.com`, has a full page dedicated to this question, which provides some excellent background material. It also offers a succinct one-liner that is worth adopting as an operational definition: **machine learning** is the training of a model from data that generalizes a decision against a performance measure.

With this definition in mind, we will require a few things in order to perform machine learning. The first is that we have the data. The second is that a pattern actually exists, which is to say that with known input values from our training data, we can make a prediction or decision based on data that we did not use to train the model. This is the **generalization** in machine learning. Third, we need some sort of performance measure to see how well we are learning/generalizing, for example, the mean squared error, accuracy, and others. We will look at a number of performance measures throughout the book.

One of the things that I find interesting in the world of machine learning are the changes in the language to describe the data and process. As such, I can't help but include this snippet from the philosopher, George Carlin:

> *"I wasn't notified of this. No one asked me if I agreed with it. It just happened. Toilet paper became bathroom tissue. Sneakers became running shoes. False teeth became dental appliances. Medicine became medication. Information became directory assistance. The dump became the landfill. Car crashes became automobile accidents. Partly cloudy became partly sunny. Motels became motor lodges. House trailers became mobile homes. Used cars became previously owned transportation. Room service became guest-room dining, and constipation became occasional irregularity.*
>
> *— Philosopher and Comedian, George Carlin*

I cut my teeth on datasets that had dependent and independent variables. I would build a model with the goal of trying to find the best fit. Now, I have labeled the instances and input features that require engineering, which will become the feature space that I use to learn a model. When all was said and done, I used to look at my model parameters; now, I look at weights.

The bottom line is that I still use these terms interchangeably and probably always will. Machine learning purists may curse me, but I don't believe I have caused any harm to life or limb.

Machine learning caveats

Before we pop the cork on the champagne bottle and rest easy that machine learning will cure all of our societal ills, we need to look at a few important considerations—caveats if you will—about machine learning. As you practice your craft, always keep these at the back of your mind. It will help you steer clear of some painful traps.

Failure to engineer features

Just throwing data at the problem is not enough; no matter how much of it exists. This may seem obvious, but I have personally experienced, and I know of others who have run into this problem, where business leaders assumed that providing vast amounts of raw data combined with the supposed magic of machine learning would solve all the problems. This is one of the reasons the first chapter is focused on a process that properly frames the business problem and leader's expectations.

Unless you have data from a designed experiment or it has been already preprocessed, raw, observational data will probably never be in a form that you can begin modeling. In any project, very little time is actually spent on building models. The most time-consuming activities will be on the engineering features: gathering, integrating, cleaning, and understanding the data. In the practical exercises in this book, I would estimate that 90 percent of my time was spent on coding these activities versus modeling. This, in an environment where most of the datasets are small and easily accessed. In my current role, 99 percent of the time in SAS is spent using PROC SQL and only 1 percent with things such as PROC GENMOD, PROC LOGISTIC, or Enterprise Miner.

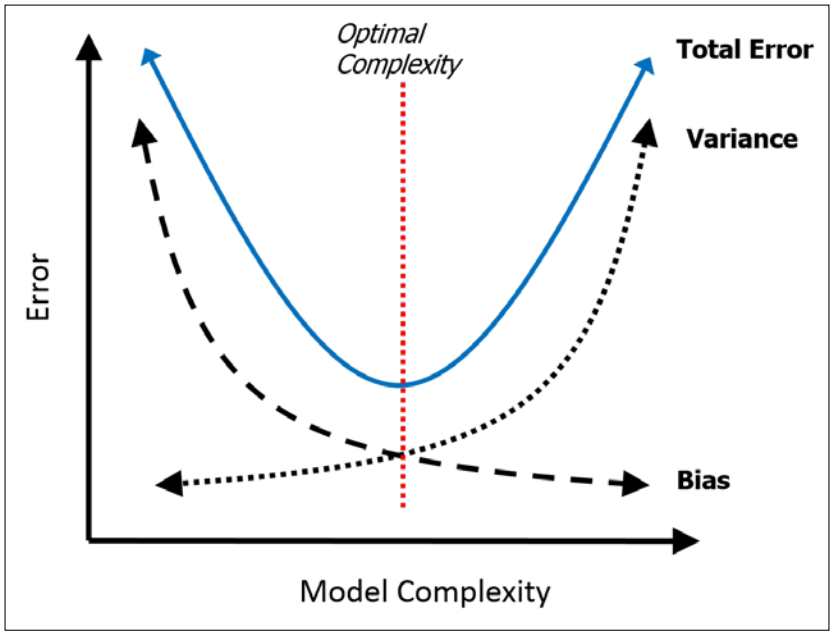
When it comes to feature engineering, I fall in the camp of those that say there is no substitute for domain expertise. There seems to be another camp that believes machine learning algorithms can indeed automate most of the feature selection/engineering tasks and several start-ups are out to prove this very thing. (I have had discussions with a couple of individuals that purport their methodology does exactly that but they were closely guarded secrets.) Let's say that you have several hundred candidate features (independent variables). A way to perform automated feature selection is to compute the univariate information value. However, a feature that appears totally irrelevant in isolation can become important in combination with another feature. So, to get around this, you create numerous combinations of the features. This has potential problems of its own as you may have a dramatically increased computational time and cost and/or overfit your model. Speaking of overfitting, let's pursue it as the next caveat.

Overfitting and underfitting

Overfitting manifests itself when you have a model that does not generalize well. Say that you achieve a classification accuracy rate on your training data of 95 percent, but when you test its accuracy on another set of data, the accuracy falls to 50 percent. This would be considered a high variance. If we had a case of 60 percent accuracy on the `train` data and 59 percent accuracy on the `test` data, we now have a low variance but a high bias. This bias-variance trade-off is fundamental to machine learning and model complexity.

Let's nail down the definitions. A bias error is the difference between the value or class that we predict and the actual value or class in our training data. A variance error is the amount by which the predicted value or class in our training set differs from the predicted value or class versus the other datasets. Of course, our goal is to minimize the total error (bias + variance), but how does that relate to model complexity?

For the sake of argument, let's say that we are trying to predict a value and we build a simple linear model with our `train` data. As this is a simple model, we could expect a high bias, while on the other hand, it would have a low variance between the `train` and `test` data. Now, let's try including polynomial terms in the linear model or build decision trees. The models are more complex and should reduce the bias. However, as the bias decreases, the variance, at some point, begins to expand and generalizability is diminished. You can see this phenomena in the following illustration. Any machine learning effort should strive to achieve the optimal trade-off between the bias and variance, which is easier said than done.



We will look at methods to combat this problem and optimize the model complexity, including cross-validation (*Chapter 2, Linear Regression - The Blocking and Tackling of Machine Learning*. through *Chapter 7, Neural Networks*) and regularization (*Chapter 4, Advanced Feature Selection in Linear Models*).

Causality

It seems a safe assumption that the proverbial correlation does not equal causation—a dead horse has been sufficiently beaten. Or has it? It is quite apparent that correlation-to-causation leaps of faith are still an issue in the real world. As a result, we must remember and convey with conviction that these algorithms are based on observational and not experimental data. Regardless of what correlations we find via machine learning, nothing can trump a proper experimental design. As Professor Domingos states:

> *If we find that beer and diapers are often bought together at the supermarket, then perhaps putting beer next to the diaper section will increase sales. But short of actually doing the experiment it's difficult to tell."*
>
> *— Domingos, P., 2012)*

In *Chapter 11*, *Time Series and Causality*, we will touch on a technique borrowed from econometrics to explore causality in time series, tackling an emotionally and politically sensitive issue.

Enough of my waxing philosophically; let's get started with using R to master machine learning! If you are a complete novice to the R programming language, then I would recommend that you skip ahead and read the appendix on using R. Regardless of where you start reading, remember that this book is about the journey to master machine learning and not a destination in and of itself. As long as we are working in this field, there will always be something new and exciting to explore. As such, I look forward to receiving your comments, thoughts, suggestions, complaints, and grievances. As per the words of the Sioux warriors: Hoka-hey! (Loosely translated it means forward together)

What this book covers

Chapter 1, *A Process for Success* - shows that machine learning is more than just writing code. In order for your efforts to achieve a lasting change in the industry, a proven process will be presented that will set you up for success.

Chapter 2, *Linear Regression - The Blocking and Tackling of Machine Learning*, provides you with a solid foundation before learning advanced methods such as Support Vector Machines and Gradient Boosting. No more solid foundation exists than the least squares linear regression.

Chapter 3, *Logistic Regression and Discriminant Analysis*, presents a discussion on how logistic regression and discriminant analysis is used in order to predict a categorical outcome.

Chapter 4, *Advanced Feature Selection in Linear Models*, shows regularization techniques to help improve the predictive ability and interpretability as feature selection is a critical and often extremely challenging component of machine learning.

Chapter 5, *More Classification Techniques – K-Nearest Neighbors and Support Vector Machines*, begins the exploration of the more advanced and nonlinear techniques. The real power of machine learning will be unveiled.

Chapter 6, *Classification and Regression Trees*, offers some of the most powerful predictive abilities of all the machine learning techniques, especially for classification problems. Single decision trees will be discussed along with the more advanced random forests and boosted trees.

Chapter 7, *Neural Networks*, shows some of the most exciting machine learning methods currently used. Inspired by how the brain works, neural networks and their more recent and advanced offshoot, Deep Learning, will be put to the test.

Chapter 8, *Cluster Analysis*, covers unsupervised learning. Instead of trying to make a prediction, the goal will focus on uncovering the latent structure of observations. Three clustering methods will be discussed: hierarchical, k-means, and partitioning around medoids.

Chapter 9, *Principal Components Analysis*, continues the examination of unsupervised learning with principal components analysis, which is used to uncover the latent structure of the features. Once this is done, the new features will be used in a supervised learning exercise.

Chapter 10, *Market Basket Analysis and Recommendation Engines*, presents the techniques that are used to increase sales, detect fraud, and improve health. You will learn about market basket analysis of purchasing habits at a grocery store and then dig into building a recommendation engine on website reviews.

Chapter 11, *Time Series and Causality*, discusses univariate forecast models, bivariate regression, and Granger causality models, including an analysis of carbon emissions and climate change.

Chapter 12, *Text Mining*, demonstrates a framework for quantitative text mining and the building of topic models. Along with time series, the world of data contains vast volumes of data in a textual format. With so much data as text, it is critically important to understand how to manipulate, code, and analyze the data in order to provide meaningful insights.

R Fundamentals, shows the syntax functions and capabilities of R. R can have a steep learning curve, but once you learn it, you will realize just how powerful it is for data preparation and machine learning.

What you need for this book

As R is a free and open source software, you will only need to download and install it from `https://www.r-project.org/`. Although it is not mandatory, it is highly recommended that you download IDE and RStudio from `https://www.rstudio.com/products/RStudio/`.

Who this book is for

If you want to learn how to use R's machine learning capabilities in order to solve complex business problems, then this book is for you. An experience with R and a working knowledge of basic statistical or machine learning will prove helpful.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows. Any command-line input or output is written as follows:

```
cor(x1, y1) #correlation of x1 and y1
[1] 0.8164205


> cor(x2, y1) #correlation of x2 and y2


[1] 0.8164205
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: Clicking the **Next** button moves you to the next screen.

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail `feedback@packtpub.com`, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files from your account at `http://www.packtpub.com` for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from `https://www.packtpub.com/sites/default/files/downloads/4527OS_ColouredImages.pdf`.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to `https://www.packtpub.com/books/content/support` and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

eBooks, discount offers, and more

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `customercare@packtpub.com` for more details.

At `www.PacktPub.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 Packt books and eBooks.

Questions

If you have a problem with any aspect of this book, you can contact us at `questions@packtpub.com`, and we will do our best to address the problem.

1
A Process for Success

"If you don't know where you are going, any road will get you there."

– Robert Carrol

"If you can't describe what you are doing as a process, you don't know what you're doing."

– W. Edwards Deming

At first glance, this chapter may seem to have nothing to do with machine learning, but it has everything to do with machine learning and specifically, its implementation and making the changes happen. The smartest people, best software, and best algorithm do not guarantee success, no matter how it is defined.

In most—if not all—projects, the key to successfully solving problems or improving decision-making is not the algorithm, but the soft, more qualitative skills of communication and influence. The problem many of us have with this is that it is hard to quantify how effective one is around these skillsets. It is probably safe to say that many of us ended up in this position because of a desire to avoid it. After all, the highly successful TV comedy *The Big Bang Theory* was built on this premise. Therefore, this chapter is to set you up for success. The intent is to provide a process, a flexible process no less, where you can become a **Change Agent**: a person who can influence and turn their insights into action without positional power. We will focus on **Cross-Industry Standard Process for Data Mining** (**CRISP-DM**). It is probably the most well-known and respected of any processes for analytical projects. Even if you use another industry process or something proprietary, there should still be a few gems in this chapter that you can take away.

I will not hesitate to say that this all is easier said than done, and without question, I'm guilty of every sin by both commission and omission that will be discussed in this chapter. With skill and some luck, you can avoid the many physical and emotional scars I've picked up over the last 10 and a half years.

Finally, we will also have a look at a flow chart (a cheat sheet) that you can use to help you identify what methodology to apply to the problem at hand.

The process

The CRISP-DM process was designed specifically for the data mining. However, it is flexible and thorough enough that it can be applied to any analytical project, whether it is predictive analytics, data science, or machine learning. Don't be intimidated by the numerous list of tasks as you can apply your judgment to the process and adapt it for any real-world situation. The following figure provides a visual representation of the process and shows the feedback loops, which facilitate its flexibility:

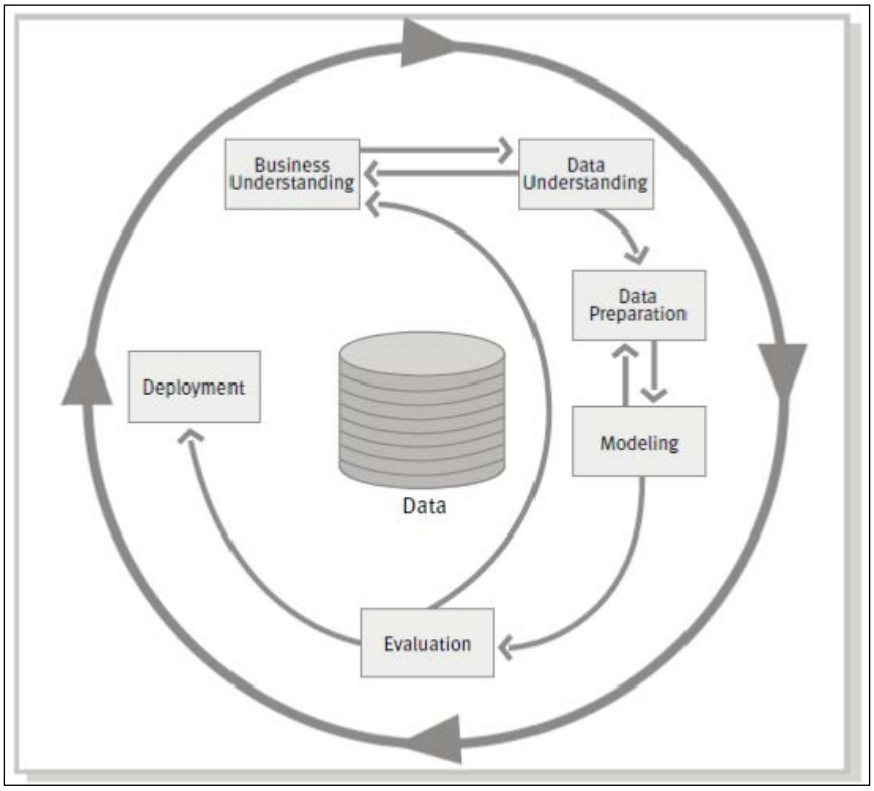


Figure from CRISP-DM 1.0, Step-by-step data mining guide

The process has the following six phases:

- **Business Understanding**
- **Data Understanding**
- **Data Preparation**
- **Modeling**
- **Evaluation**
- **Deployment**

For an in-depth review of the entire process with all of its tasks and subtasks, you can examine the paper by SPSS, CRISP-DM 1.0, step-by-step data mining guide, available at `https://the-modeling-agency.com/crisp-dm.pdf`.

I will discuss each of the steps in the process, covering the important tasks. However, it will not be in the detailed level of the guide, but more high level. We will not skip any of the critical details but focus more on the techniques that one can apply to the tasks. Keep in mind that the process steps will be used in the later chapters as a framework in the actual application of the machine learning methods in general and the R code specifically.

Business understanding

One cannot underestimate how important this first step of the process is in achieving success. It is the foundational step and failure or success here will likely determine failure or success for the rest of the project. The purpose of this step is to identify the requirements of the business so that you can translate them into analytical objectives. It has the following four tasks:

1. Identify the business objective
2. Assess the situation
3. Determine the analytical goals
4. Produce a project plan

Identify the business objective

The key to this task is to identify the goals of the organization and frame the problem. An effective question to ask is, what are we going to do different? This may seem like a benign question, but it can really challenge people to ponder what they need from an analytical perspective and it can get to the root of the decision that needs to be made. It can also prevent you from going out and doing a lot of unnecessary work on some fishing expedition. As such, the key for you is to identify the **decision**. A working definition of a decision can be put forward to the team as the irrevocable choice to commit or not commit the resources. Additionally, remember that the choice to do nothing different is indeed a decision.

This does not mean that a project should not be launched if the choices are not absolutely clear. There will be times when the problem is not or cannot be well-defined; to paraphrase former Defense Secretary Donald Rumsfeld, there are known – unknowns. Indeed, there will probably be many times when the problem is ill-defined and the project's main goal is to further the understanding of the problem and generate hypotheses; again calling on Secretary Rumsfeld, unknown – unknowns, which means that you don't know what you don't know. However, in ill-defined problems, one should go forward with an understanding of what will happen next in terms of resource commitment based on the various outcomes of hypothesis exploration.

Another thing to consider in this task is to manage expectations. There is no such thing as a perfect data, no matter what its depth and breadth is. This is not the time to make guarantees but to communicate what is possible, given your expertise.

I recommend a couple of outputs from this task. The first is a mission statement. This is not the touchy-feely mission statement of an organization, but it is your mission statement or, more importantly, the mission statement approved by the project sponsor. I stole this idea from my years of military experience and I could write volumes on why it is effective, but that is for another day. Let's just say that in the absence of clear direction or guidance, the mission statement or whatever you want to call it becomes the unifying statement and can help prevent scope creep. It consists of the following points:

- **Who**: This is yourself or the team or project name; everyone likes a cool project name, for example, Project Viper, Project Fusion, and so on
- **What**: This is the task that you will perform, for example, conduct machine learning
- **When**: This is the deadline
- **Where**: This could be geographical; by function, department, initiative, and so on
- **Why**: This is the purpose of doing the project, that is, the business goal

The second task is to have as clear a definition of success as possible. Literally, ask what does success look like? Help the team/sponsor paint a picture of success that you can understand. Your job then is to translate this into modeling requirements.

Assess the situation

This task helps you in project planning by gathering information on the resources available, constraints, and assumptions, identifying the risks, and building contingency plans. I would further add that this is also the time to identify the key stakeholders that will be impacted by the decisions to be made.

A couple of points here. When examining the resources that are available, do not neglect to scour the records of the past and current projects. Odds are someone in the organization has or is working on the same problem and it may be essential to synchronize your work with theirs. Don't forget to enumerate the risks considering time, people, and money. Do everything in your power to create a list of the stakeholders, both those that impact your project and those that could be impacted by your project. Identify who these people are and how they can influence/be impacted by the decision. Once this is done, work with the project sponsor to formulate a communication plan with these stakeholders.

Determine the analytical goals

Here, you are looking to translate the business goal into technical requirements. This includes turning the success criterion from the task of creating a business objective to technical success. This might be things such as RMSE or a level of predictive accuracy.

Produce a project plan

The task here is to build an effective project plan with all the information gathered up to this point. Regardless of what technique you use, whether it be a Gantt chart or some other graphic, produce it and make it a part of your communication plan. Make this plan widely available to the stakeholders and update it on a regular basis and as circumstances dictate.

Data understanding

After enduring the all-important pain of the first step, you can now get your hands on the data. The tasks in this process consist of the following:

1. Collect the data
2. Describe the data
3. Explore the data
4. Verify the data quality

This step is the classic case of **ETL** is **Extract**, **Transform**, **Load**. There are some considerations here. You need to make an initial determination that the data available is adequate to meet your analytical needs. As you explore the data, visually and otherwise, determine if the variables are sparse and identify the extent to which the data may be missing. This may drive the learning method that you use and/or whether the imputation of the missing data is necessary and feasible.

Verifying the data quality is critical. Take the time to understand who collects the data, how it is collected, and even why it is collected. It is likely that you may stumble upon an incomplete data collection, cases where unintended IT issues led to errors in the data, or there were planned changes in the business rules. This is critical in the time series where often business rules change over time on how the data is classified. Finally, it is a good idea to begin documenting any code at this step. As a part of the documentation process, if a data dictionary is not available, save yourself the heartache later on and make one.

Data preparation

Almost there! This step has the following five tasks:

1. Select the data
2. Clean the data
3. Construct the data
4. Integrate the data
5. Format the data

These tasks are relatively self-explanatory. The goal is to get the data ready to input in the algorithms. This includes merging, feature engineering, and transformations. If imputation is needed, then it happens here as well. Additionally, with R, pay attention to how the outcome needs to be labeled. If your outcome/response variable is Yes/No, it may not work in some packages and will require a transformed or no variable with 1/0. At this point, you should also break your data into the various test sets if applicable: train, test, or validate. This step can be an unforgivable burden, but most experienced people will tell you that it is where you can separate yourself from your peers. With this, let's move on to the money step.

Modeling

This is where all the work that you've done up to this point can lead to fist-pumping exuberance or fist-pounding exasperation. But hey, if it was that easy, everyone would be doing it. The tasks are as follows:

1. Select a modeling technique
2. Generate a test design
3. Build a model
4. Assess a model

Oddly, this process step includes the considerations that you have already thought of and prepared for. In the first step, one will need at least a modicum of an idea about how they will be modeling. Remember, that this is a flexible, iterative process and not some strict linear flowchart such as an aircrew checklist.

The cheat sheet included in this chapter should help guide you in the right direction for the modeling techniques. A test design refers to the creation of your test and train datasets and/or the use of cross-validation and this should have been thought of and accounted for in the data preparation.

Model assessment involves comparing the models with the criteria/criterion that you developed in the business understanding, for example, RMSE, Lift, ROC, and so on.

Evaluation

With the evaluation process, the main goal is to confirm that the work that has been done and the model selected at this point meets the business objective. Ask yourself and others, have we achieved the definition of success? Let the Netflix prize serve as a cautionary tale here. I'm sure you are aware that Netflix awarded a $1 million prize to the team that could produce the best recommendation algorithm as defined by the lowest RMSE. However, Netflix did not implement it because the incremental accuracy gained was not worth the engineering effort! Always apply Occam's razor. At any rate, here are the tasks:

1. Evaluate the results
2. Review the process
3. Determine the next steps

In reviewing the process, it may be necessary—as you no doubt determined earlier in the process—to take the results through governance and communicate with the other stakeholders in order to gain their buy-in. As for the next steps, if you want to be a change agent, make sure that you answer the **what**, **so what**, and **now what** in the stakeholders' minds. If you can tie their now what into the decision that you made earlier, you are money.

Deployment

If everything is done according to the plan up to this point, it might just come down to flipping a switch and your model goes live. Assuming that this is not the case, here are the tasks of this step:

1. Deploying the plan
2. Monitoring and maintenance of the plan
3. Producing the final report
4. Reviewing the project

After the deployment and monitoring/maintenance is underway, it is crucial for yourself and those that will walk in your steps to produce a well-written final report. This report should include a white paper and briefing slide. I have to say that I resisted the drive to put my findings in a white paper as I was an indentured servant to the military's passion for PowerPoint slides. However, slides can and will be used against you, cherry-picked or misrepresented by various parties for their benefit. Trust me, that just doesn't happen with a white paper as it becomes an extension of your findings and beliefs.

Now for the all-important process review. You may have your own proprietary way of conducting it, but here is what it should cover, whether you conduct it in a formal or informal way:

- What was the plan?
- What actually happened?
- Why did it happen or did not happen?
- What should be sustained in future projects?
- What should be improved upon in future projects?
- Create an action plan to ensure sustainment and improvement happens

That concludes the review of the CRISP-DM process, which provides a comprehensive and flexible framework to guarantee the success of your project and make you an agent of change.

Algorithm flowchart

The purpose of this section is to create a tool that will help you not just select the possible modeling techniques but also to think deeper about the problem. The residual benefit is that it may help you frame the problem with the project sponsor/team. The techniques in the flowchart are certainly not comprehensive but are exhaustive enough to get you started. It also includes techniques not discussed in this book.

The following figure starts the flow of selecting the potential modeling techniques. As you answer the question(s), it will take you to one of the four additional charts:

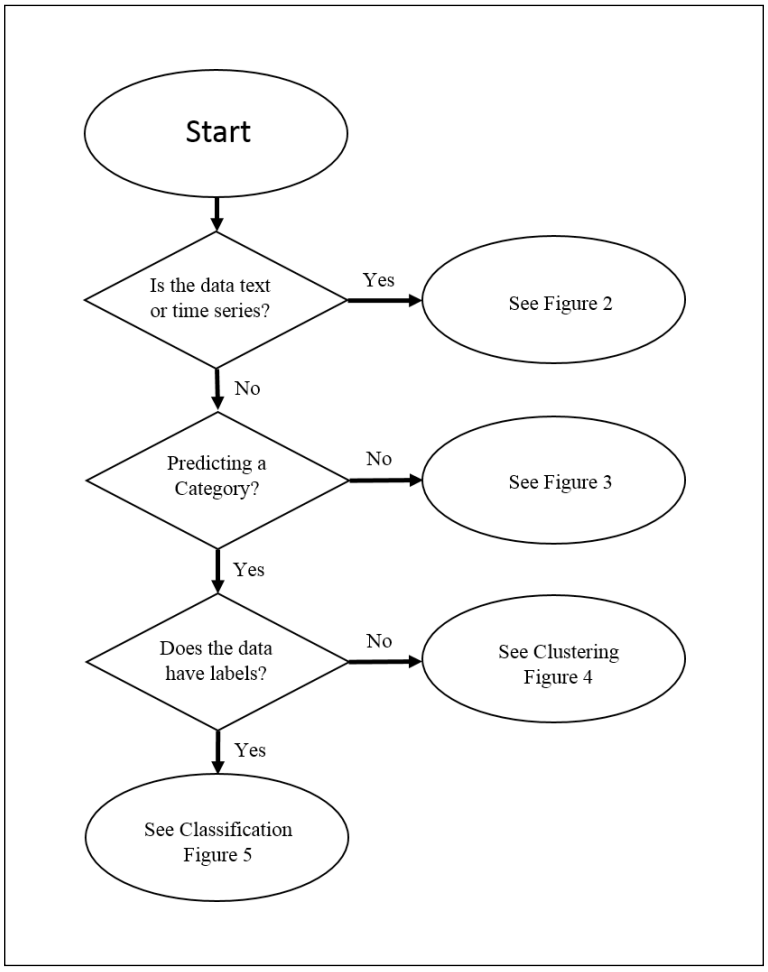


Figure 1

If the data is a text or in the time series format, then you will follow the flow in the following figure:

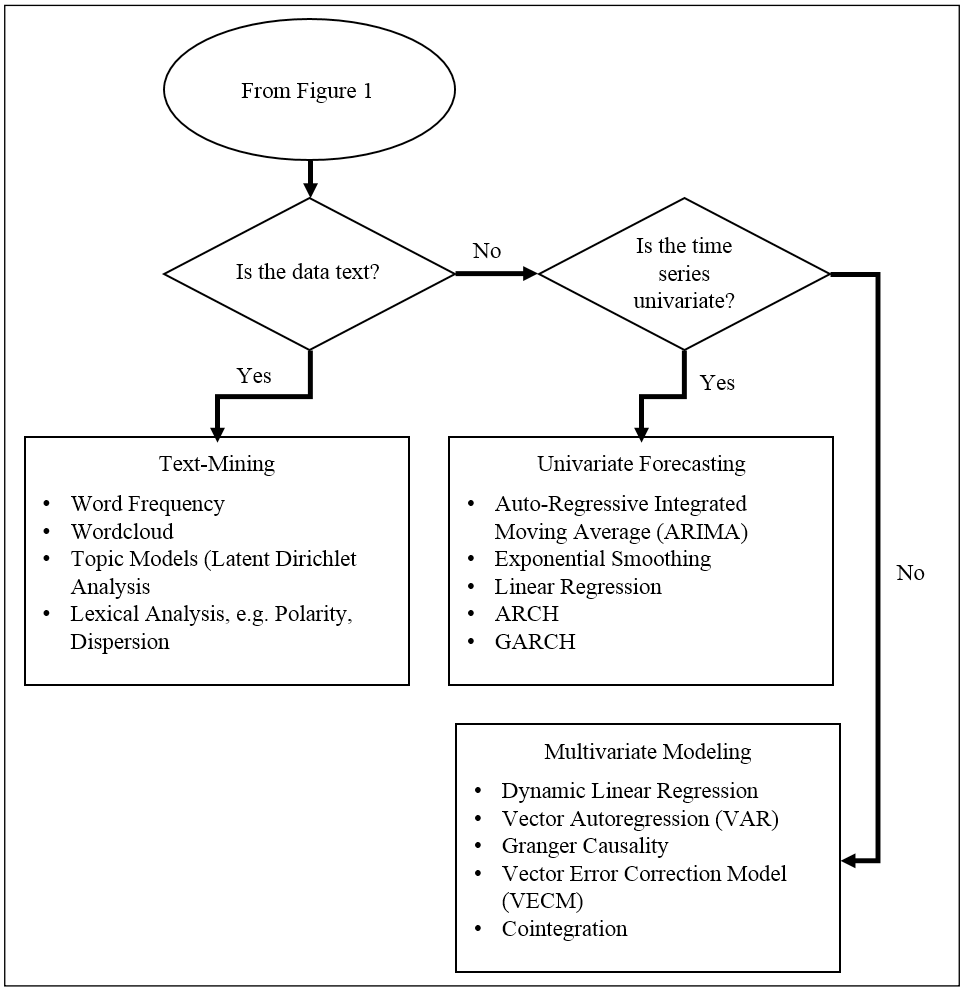


Figure 2

In this branch of the algorithm, you do not have a text or the time series data. Additionally, you are not trying to predict what category the observations belong to.

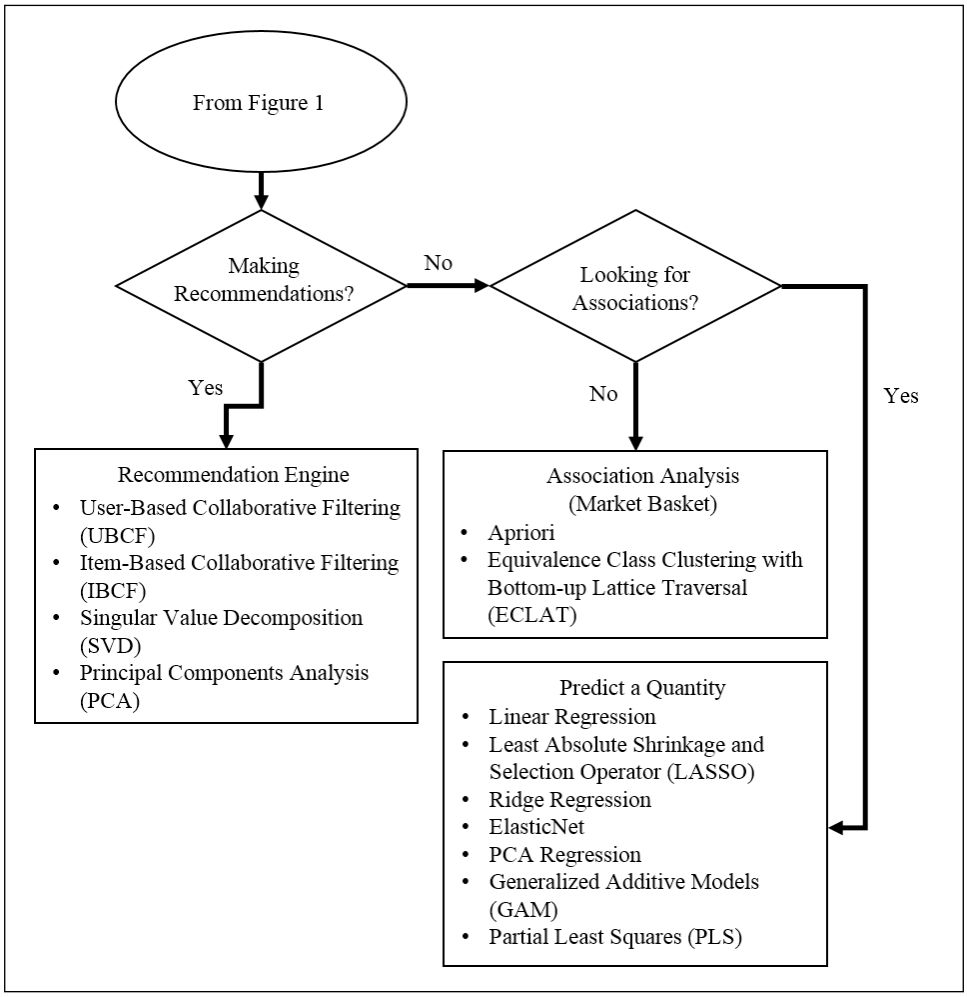


Figure 3

To get to this section, you would have data that is not text or time series. You want to categorize the data, but it does not have an outcome label, which brings us to clustering methods, as follows:

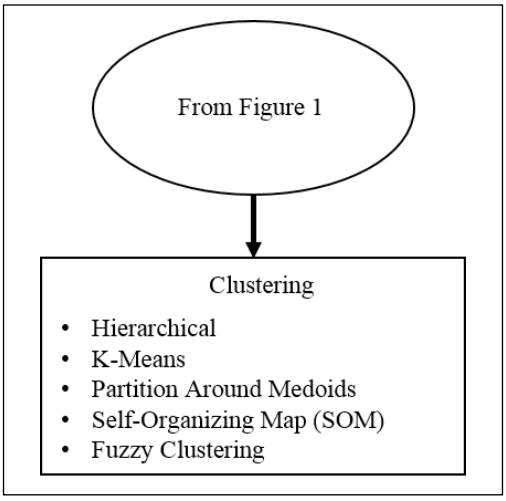


Figure 4

This brings us to a situation where we want to categorize the data and it is labeled, that is, classification:

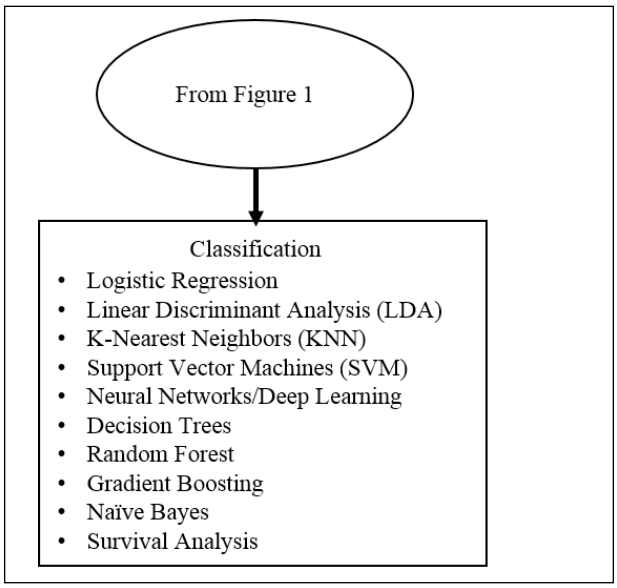


Figure 5

Summary

This chapter was about how to set yourself and your team up for success in any project that you tackle. The CRISP-DM process is put forward as a flexible and comprehensive framework in order to facilitate the softer skills of communication and influence. Each process step and the tasks in each step were enumerated. More than that, the commentary provides some techniques and considerations to help in the process execution. By taking heed of the process, you can indeed become an agent of positive change to any organization.

The other item put forth in this chapter was an algorithm flowchart; a cheat sheet to help in identifying the proper techniques to apply in order to solve the business problem. With this foundation in place, we can now move on to applying these techniques to real-world problems.

2
Linear Regression – The Blocking and Tackling of Machine Learning

"Some people try to find things in this game that don't exist, but football is only two things – blocking and tackling."

– Vince Lombardi, Hall of Fame Football Coach

It is important that we get started with a simple, yet extremely effective, technique that has been used for a long time: **linear regression**. Albert Einstein is believed to have remarked at one time or another that things should be made as simple as possible, but no simpler. This is sage advice and a good rule of thumb in the development of algorithms for machine learning. Considering the other techniques that we will discuss later, there is no simpler model than the tried and tested linear regression, which uses the **least squares approach** to predict a quantitative outcome. In fact, one could consider it to be the foundation of all the methods that we will discuss later, many of which are mere extensions. If you can master the linear regression method, well, then quite frankly, I believe you can master the rest of this book. Therefore, let us consider this a good point for starting start our journey towards becoming a machine-learning guru.

This chapter covers introductory material, and an expert in this subject can skip ahead to the next topic. Otherwise, ensure that you thoroughly understand this topic before venturing on to other, more complex learning methods. I believe you will discover that many of your projects can be addressed by just applying what is discussed in the following section. Linear regression is probably the easiest model to explain to your customers, most of whom will have at least a cursory understanding of **R-squared**. Many of them will have been exposed to it at great depth and thus, be comfortable with variable contribution, **collinearity**, and the like.