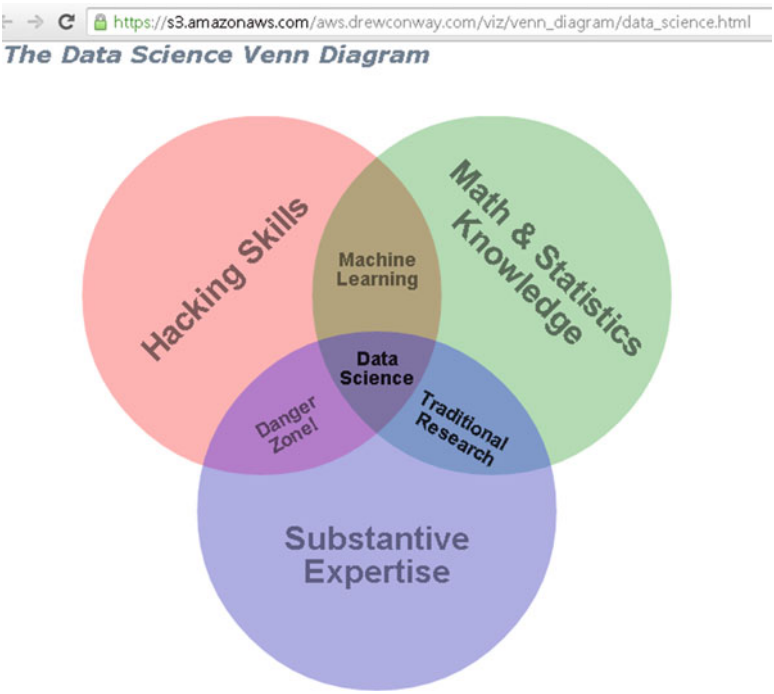
Chapter 2
An Approach for Data Scientists

What is a data scientist? A data scientist is one who had inter-disciplinary skills in both programming, statistics, and business domains to create actionable insights based on experiments or summaries from data. One of the most famous definitions is from Drew Conway.



A Ohri, *R for Cloud Computing*, DOI 10.1007/978-1-4939-1702-0_2

On a daily basis, a data scientist is simply a person

- who can write some code
  - in one or more of the languages of R, Python, Java, SQL, Hadoop (Pig, HQL, MR)
- for
  - data storage, querying, summarization, visualization efficiently, and in time
- on
  - databases, on cloud, servers, and understand enough statistics to derive insights from data so business can make decisions

What should a data scientist know? He should know how to get data, store it, query it, manage it, and turn it into actionable insights. The following approach elaborates on this simple and sequential premise.

2.1 Where to Get Data?

A data scientist needs data to do science on, right! Some of the usual sources of data for a data scientist are:

APIs—API is an acronym for Application Programming Interface. We cover APIs in detail in Chap. 7. APIs is how the current Big Data paradigm is enabled, as it enables machines to talk and fetch data from each other programmatically. For a list of articles written by the same author on APIs—see https://www.programmableweb.com/profile/ajayohri.

Internet Clickstream Logs—Internet clickstream logs refer to the data generated by humans when they click specific links within a webpage. This datum is time stamped, and the uniqueness of the person clicking the link can be established by IP address. IP addresses can be parsed by registries like https://www.arin.net/whois or http://www.apnic.net/whois for examining location (country and city), internet service provider, and owner of the address (for website owners this can be done using the website http://who.is/). In Windows using the command ipconfig and in Linux systems using ifconfig can help us examine IP Address. You can read this for learning more on IP addresses http://en.wikipedia.org/wiki/IP_address. Software like Clicky from (http://getclicky.com) and Google Analytics(www.google.com/analytics) also helps us give data which can then be parsed using their APIs. (See https://code.google.com/p/r-google-analytics/ for Google Analytics using R).

Machine Generated Data—Machines generate a lot of data especially for sensors to ensure that the machine is working properly. This datum can be logged and can be used with events like cracks or failures to have predictive asset maintenance of M2M (Machine to Machine) Analytics.

Surveys—Surveys are mostly questionnaires filled by humans. They used to be administered manually over paper, but online surveys are now the definitive trend. Surveys reveal valuable data about current preferences of current and potential customers. They do suffer from the bias inherent from design of questions by the creator. Since customer preferences evolve surveys help in getting primary data about current preferences. Coupled with stratified random sampling, they can be a powerful method for collecting data. SurveyMonkey is one such company that helps create online questionnaires (https://www.surveymonkey.com/pricing/).

Commercial Databases—Commercial Databases are proprietary databases that have been collected over time and are sold/rented by vendors. They can be used for prospect calling, appending information to existing database, and refining internal database quality.

Credit Bureaus—Credit bureaus collect financial information about people, and this information is then available for marketing organizations (subject to legal and privacy guidelines). The cost of such information is balanced by the added information about customers.

Social Media—Social media is a relatively new source of data and offers powerful insights albeit through a lot of unstructured data. Companies like Datasift offer social media data, and companies like Salesforce/Radian6 offer social media tools (http://www.salesforcemarketingcloud.com/). Facebook has 829 million daily active users on average in June 2014 with 1.32 billion monthly active users. Twitter has 255 million monthly active users and 500 million Tweets are sent per day. That generates a lot of data about what current and potential customers are thinking and writing about your products.

2.2 Where to Process Data?

Now you have the data. We need computers to process it.

- Local Machine—Benefits of storing the data in local machine are ease of access. The potential risks include machine outages, data recovery, data theft (especially for laptops) and limited scalability. A local machine is also much more expensive in terms of processing and storage and gets obsolete within a relatively short period of time.
- Server—Servers respond to requests across networks. They can be thought of as centralized resources that help cut down cost of processing and storage. They can be an intermediate solution between local machines and clouds, though they have huge capital expenditure upfront. Not all data that can fit on a laptop should be stored on a laptop. You can store data in virtual machines on your server and connected through thin shell clients with secure access.
- Cloud—The cloud can be thought of a highly scalable, metered service that allows requests from remote networks. They can be thought of as a large bank of servers but that is a simplistic definition. The more comprehensive definition is given in Chapter 1.

This chapter and in fact this book holds that the cloud approach benefits data scientists the most because it is cheap, secure, and easily scalable. The only hindrance to adoption to the cloud is conflict within existing IT department whose members are not trained to transition and maintain the network over cloud as they used to do for enterprise networks.

2.2.1 Cloud Processing

We expand on the cloud processing part.

- **Amazon** EC2—Amazon Elastic Compute Cloud (Amazon EC2) provides scalable processing power in the cloud. It has a web-based management console, has a command line tool, and offers resources for Linux and Windows virtual images. Further details are available at http://aws.amazon.com/ec2/. Amazon EC2 is generally considered the industry leader. For beginners a 12-month basic preview is available for free at http://aws.amazon.com/free/ that can allow practitioners to build up familiarity.
- **Google** Compute—https://cloud.google.com/products/compute-engine/
- **Microsoft** Azure—https://azure.microsoft.com/en-us/pricing/details/virtual-machines/ Azure Virtual Machines enable you to deploy a Windows Server, Linux, or third-party software images to Azure. You can select images from a gallery or bring your own customized images. Charge for Virtual Machines is by the minute. Discounts can range from 205 to 32 % depending if you prepay 6 months or 12 month plans and based on usage tier.
- **IBM** shut down its SmartCloud Enterprise cloud computing platform by Jan. 31, 2014 and will migrate those customers to its SoftLayer cloud computing platform, which was an IBM acquired company https://www.softlayer.com/virtual-servers
- **Oracle** Oracle's plans for the cloud are still in preview for enterprise customers as of July 2014. https://cloud.oracle.com/compute

2.3 Where to Store Data?

We need to store data in a secure and reliable environment for speedy and repeated access. There is a cost of storing this data, and there is a cost of losing the data due to some technical accident.

You can store data in the following way

- csv files, spreadsheet, and text files locally especially for smaller files. Note while this increases ease of access, it also creates problems of version control as well as security of confidential data.
- relational databases (RDBMS) and data warehouses

- hadoop based storage
- noSQL databases—These are the next generation of databases. They are non-relational, distributed, open-source, and horizontally scalable. A complete list of NoSQL databases is at http://nosql-database.org/. Notable NoSQL databases are MongoDB, couchDB, etc.
  - key-value store—Key-value stores use the map or dictionary as their fundamental data model. In this model, data is represented as a collection of key-value pairs, such that each possible key appears at most once in the collection.
    - Redis—Redis is an open source, BSD licensed, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets, and sorted sets (http://redis.io/). Redis cloud is a fully managed cloud service for hosting and running your redis dataset (http://redislabs.com/redis-cloud). rredis: An R package for the Redis persistent key-value database available from http://redis.io.http://cran.r-project.org/web/packages/rredis/index.html
    - Riak is an open source, distributed database. http://basho.com/riak/. Riak CS (Cloud Storage) is simple, open source storage software built on top of Riak. It can be used to build public or private clouds
    - MemcacheDB is a persistence enabled variant of memcached, a general-purpose distributed memory caching system often used to speed up dynamic database-driven websites by caching data and objects in memory. The main difference between MemcacheDB and memcached is that MemcacheDB has its own key-value database system based on Berkeley DB, so it is meant for persistent storage rather than as a cache solution
- column oriented databases
- cloud storage

2.3.1 Cloud Storage

- Amazon—Amazon Simple Storage Services (S3)—Amazon S3 provides a simple web-services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. http://aws.amazon.com/s3/. Cost is a maximum of 3 cents per GB per month. There are three types of storage: Standard Storage, Reduced Redundancy Storage, Glacier Storage. Reduced Redundancy Storage (RRS) is a storage option within Amazon S3 that enables customers to reduce their costs by storing non-critical, reproducible data at lower levels of redundancy than Amazon S3 standard storage. Amazon Glacier stores data for as little as $0.01 per gigabyte per month and is optimized for data that is infrequently accessed and for which retrieval times of 3 to 5 hours are suitable. These details can be seen at http://aws.amazon.com/s3/pricing/

- Google—Google Cloud Storage https://cloud.google.com/products/cloud-storage/. It also has two kinds of storage. Durable Reduced Availability Storage enables you to store data at lower cost, with the tradeoff of lower availability than standard Google Cloud Storage. Prices are 2.6 cents for Standard Storage (GB/Month) and 2 cents for Durable Reduced Availability (DRA) Storage (GB/Month). They can be seen at https://developers.google.com/storage/pricing#storage-pricing.
- Azure—Microsoft has different terminology for its cloud infrastructure. Storage is classified into three types with a fourth type (Files) being available as a preview. There are three levels of redundancy: Locally Redundant Storage (LRS),Geographically Redundant Storage (GRS), Read-Access Geographically Redundant Storage (RA-GRS): You can see details and prices at https://azure.microsoft.com/en-us/pricing/details/storage/.
- Oracle Storage is available at https://cloud.oracle.com/storage and costs around 30$ / TB per month.

2.3.2 Databases on the Cloud

- Amazon
  - Amazon RDS—Managed MySQL, Oracle, and SQL Server databases. http://aws.amazon.com/rds/. While relational database engines provide robust features and functionality, scaling a workload beyond a single relational database instance is highly complex and requires significant time and expertise.
  - DynamoDB—Managed NoSQL database service. http://aws.amazon.com/dynamodb/. Amazon DynamoDB focuses on providing seamless scalability and fast, predictable performance. It runs on solid state disks (SSDs) for low-latency response times, and there are no limits on the request capacity or storage size for a given table. This is because Amazon DynamoDB automatically partitions your data and workload over a sufficient number of servers to meet the scale requirements you provide.
  - Redshift—It is a managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyse all your data using your existing business intelligence tools. You can start small for just $0.25 per hour and scale to a petabyte or more for $1,000 per terabyte per year. http://aws.amazon.com/redshift/.
  - SimpleDB—It is highly available and flexible non-relational data store that offloads the work of database administration. Developers simply store and query data items via web services requests http://aws.amazon.com/simpledb/. A table in Amazon SimpleDB has a strict storage limitation of 10 GB and is limited in the request capacity it can achieve (typically under 25 writes/second); it is up to you to manage the partitioning and re-partitioning of

your data over additional SimpleDB tables if you need additional scale. While SimpleDB has scaling limitations, it may be a good fit for smaller workloads that require query flexibility. Amazon SimpleDB automatically indexes all item attributes and thus supports query flexibility at the cost of performance and scale.

- Google
  - Google Cloud SQL—Relational Databases in Google's Cloud https://developers.google.com/cloud-sql/
  - Google Cloud Datastore—Managed NoSQL Data Storage Service https://developers.google.com/datastore/
  - Google Big Query—Enables you to write queries on huge datasets. BigQuery uses a columnar data structure, which means that for a given query, you are only charged for data processed in each column, not the entire table https://cloud.google.com/products/bigquery/.
- Databases—This is expanded more in Chap. 6
  - RDBMS
  - Document DBs
  - Monet DB
  - Graph Databases
- How to query data?
  - SQL—we can use SQL within R using the sqldf package.
  - Pig
  - Hive QL
- Prepackaged R packages or Python Libraries that do the job:
  - AWS.tools: R package to use Amazon Web Services (http://cran.r-project.org/web/packages/AWS.tools/index.html)
  - The bigrquery provides a read-only interface to Google BigQuery. It makes it easy to retrieve metadata about your projects, datasets, tables and jobs, and provides a convenient wrapper for working with bigquery from R. (https://github.com/hadley/bigrquery)

2.4 Basic Statistics for Data Scientists

Some of the basic statistics that every data scientist should know are given here. This assumes rudimentary basic knowledge of statistics (like measures of central tendency or variation) and basic familiarity with some of the terminology used by statisticians.

- Random Sampling—In truly random sampling, the sample should be representative of the entire data. Random sampling remains of relevance in the era of Big Data and Cloud Computing
- Distributions—A data scientist should know the distributions (normal, Poisson, Chi Square, F) and also how to determine the distribution of data.
- Hypothesis Testing—Hypothesis testing is meant for testing assumptions statistically regarding values of central tendency (mean, median) or variation. A good example of an easy-to-use software for statistical testing is the "test" tab in the Rattle GUI.
- Outliers—Checking for outliers is a good way for a data scientist to see anomalies as well as identify data quality. The box plot (exploratory data analysis) and the outlierTest function from car package (Bonferroni Outlier Test) are how statistical rigor can be maintained to outlier detection.

2.5 Basic Techniques for Data Scientists

Some of the basic techniques that a data scientist must know are listed as follows:

- Text Mining—In text mining, text data are analysed for frequencies, associations, and correlation for predictive purposes. The tm package from R greatly helps with text mining.
- Sentiment Analysis—In sentiment analysis the text data are classified based on a sentiment lexicography (e.g., which says happy is less positive than delighted but more positive than sad) to create sentiment scores of the text data mined.
- Social Network Analysis—In social network analysis, the direction of relationships, the quantum of messages, and the study of nodes, edges, and graphs is done to give insights.
- Time Series Forecasting—Data is said to be auto regressive with regards to time if a future value is dependent on a current value for a variable. Techniques such as ARIMA and exponential smoothing and R packages like forecast greatly assist in time series forecasting.
- Web Analytics
- Social Media Analytics

2.6 How to Store Output?

Congratulations, you are done with your project! How do we share it so that both code and documentation are readable and reproducible. While the traditional output was a powerpoint, a spreadsheet, or a word document, some of the newer forms of output are.

- Markdown—a minimalistic form of HTML.
- You can read the cheat sheet on Markdown at http://shiny.rstudio.com/articles/rm-cheatsheet.html
  - HTML 5—see the slidify package
  - Latex—see sweave (for typesetting)

2.7 How to Share Results and Promote Yourself?

Data scientists are known when they publish their results internally and externally of their organization with adequate data hygiene safeguards. Some of the places on the internet to share your code or results are:

- github
  - Rpubs
  - meetups
  - twitter
  - blogs and linkedin

2.8 How to Move or Query Data?

We use utilities like wget, winscp, or curl to move massive amounts of data.

2.8.1 Data Transportation on the Cloud

This typically referred to as Network Costs or Ingress/Egress costs for data, and these can be critical components for costing when large amounts of data are being transferred. While traditional approach was ETL for data warehouses, we can simplify the data transportation costs on the cloud, extract, transform, and load

ETL(extract, transform, and load) refers to a process in database usage and especially in data warehousing that: extracts data from outside sources, transforms it to fit operational needs, which can include quality levels, loads it into the end target (database, more specifically, operational data store, data mart, or data warehouse). We will cover more on databases in Chap. 6.

2.9 How to Analyse Data?

The concept of Garbage In, Garbage Out (GiGo) stems from the fact that computers can only produce output that is dependent on quality of input. So data quality is an important factor for the production of quality analysis. The concept of tidy data further helps us in gauging and refining data quality.

2.9.1 Data Quality—Tidy Data

The most common problems with messy datasets

1. column headers are values, not variable names
2. multiple variables are stored in one column
3. variables are stored in both rows and columns
4. multiple types of observational units are stored in the same table
5. a single observational unit is stored in multiple tables

In tidy data:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

For more on tidy data, refer to the paper by Hadley Wickham in "References" section of this chapter. In addition a new package called tidyr has been introduced.

You can read about this new package at http://blog.rstudio.org/2014/07/22/introducing-tidyr/

tidyr is new package that makes it easy to "tidy" your data. Tidy data is data that is easy to work with: it is easy to munge (with dplyr), visualize (with ggplot2 or ggvis), and model (with R's hundreds of modeling packages).

tidyr provides three main functions for tidying your messy data: gather(), separate() and spread().

- gather() takes multiple columns, and gathers them into key-value pairs: it makes "wide" data longer.
- Sometimes two variables are clumped together in one column. separate() allows you to tease them apart

2.9.2 Data Quality—Treatment

The following treatments are done on data to improve quality.

1) **Missing Value Treatment** deals with missing and incomplete data. This is usually done by deletion, imputation by (mean, median, or correlation) or creation of a categorical value (missing value flag) which indicates whether value was missing or not.
2) **Outlier Treatment** deals with extreme values of data, usually by capping the maximum and creating a minimum. For example age of adults may be capped at 20 years minimum, and 80 years maximum and any values below 20 may be imputed to be rounded off to 20, and any data indicating age above 80 may be capped at 80.

2.9.3 Data Quality—Transformation

The following transformations can help in creating new variables that make more coherent sense than doing analysis on original data. If the input data is X, the transformed value can be X^2 (square), X ^0.5 (Square Root), log(X), exp(X), and 1/X (inverse).

2.9.4 Split Apply Combine

In the split-apply-combine strategy you split the data into smaller number of parts, you apply a function to it, then you combine these parts together. It is computationally easier than applying the function to the whole data at once. The R package plyr is based on this and greatly helps with simplifying data structures. For more on split apply combine see the *Journal of Statistical Software* paper in the "References" part of this chapter.

2.10 How to Manipulate Data?

We need to bring data into the desired shape, dimension so as to analyse and represent it.

Some of the standard ways of manipulating data are:

- Grouping
- Selecting
- Conditional Selecting
- Transforming
- Slicing and Dicing Data

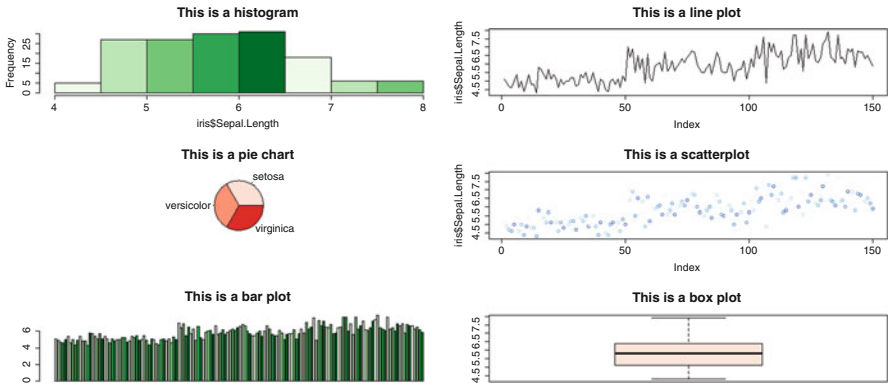
The book Data Manipulation with R by Phil Spector deals more with this topic. http://www.springer.com/mathematics/probability/book/978-0-387-74730-9. You can also see the R Tutorial in Chap. 5 for basic data manipulation.

2.10.1 Data Visualization

Data visualization is a new branch of descriptive statistics. Some of the points to remember in this topic are:

Grammar of Graphics was a book written by Leland Wilkinson and implemented in R by Dr Hadley Wickham for the ggplot2 package. The ggplot2 package gives considerable aesthetic lift over native graphics in R. Basically a graph is considered a mix of axis, plot type, statistical transformation, and data.

Types of Graphs—The following graph shows the basic types of graphs. Most data scientists are expected to know basic as well as advanced graphs. A good guide is available at http://www.statmethods.net/graphs/.



2.11 Project Methodologies

Knowledge discovery in databases (KDD) is the process of (semi-)automatic extraction of knowledge from databases which is valid, previously unknown, and potentially useful. Project methodologies help to break down complex projects into manageable chunks. The three most commonly used methodologies are:

- DMAIC—This is expanded as Define, Measure, Analyse, Improve and Control and is part of the six sigma methodology for sustained quality improvement. Define the problem needs to be solved, measure the capability of the process, analyse when and where do defects occur, improve process capability by

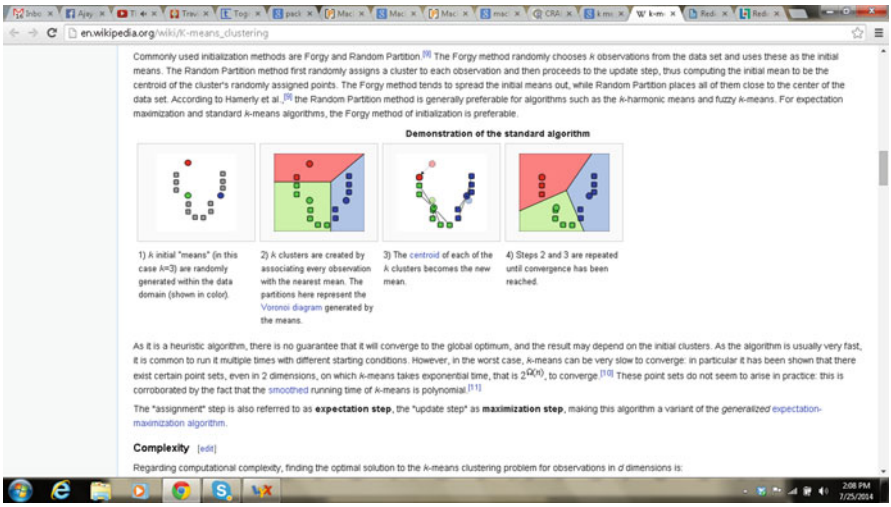
reducing the process variation and looking at the vital factors, and lastly by putting in place controls to sustain the gains.

- SEMMA—This stands for Sample, Explore, Modify, Model, and Assess. It was developed by the SAS Institute and is considered a logical and sequential methodology for carrying out data mining.
- CRIPS-DM—Cross Industry Standard Process for Data Mining was created by ESPRIT from European Union. Its constituents and methodology can be shown as follows.

2.12 Algorithms for Data Science

An **algorithm** is simply a step by step procedure for calculations. It is thus a procedure or formula for solving a problem. Some of the most widely used algorithms by data scientists are

- **Kmeans (Clustering)—**An illustration is shown below.



- **Ordinary Least Squares (Regression)**—This method minimizes the sum of squared-vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation.
- **Apriori Algorithm** is used in Association Rules and Market Basket Analysis to find which combination of transactions appears most frequently together over a minimum threshold support. A good example and demonstration is given in the "References" part of this chapter.
- **K Nearest Neighbours**—An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour.
- **Support Vector Machines**—An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.
- **Naive Bayes** classifier—Bayes theorem says The probability P(A|B) of "A assuming B" is given by the formula P(A|B) = P(AB) / P(B). A naive Bayes classifier assumes that the value of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3″ in diameter.
- **Neural Nets**—These were inspired by how nature and human brain works. An example system has three layers. The first layer has input neurons which send data via synapses to the second layer of neurons, and then via more synapses to

the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations.

For a more exhaustive list see http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R. It has the following chapters and algorithms covered.

- **Dimensionality Reduction**—Principal Component Analysis, Singular Value Decomposition, Feature Selection
- **Frequent Pattern Mining**—The Eclat Algorithm, arulesNBMiner, The Apriori Algorithm, The FP-Growth Algorithm
- **Sequence Mining**—SPADE, DEGSeq
- **Clustering**—K-Means, Hybrid Hierarchical Clustering, Expectation Maximization (EM),Dissimilarity Matrix Calculation,Hierarchical Clustering,Bayesian Hierarchical Clustering,Density-Based Clustering, K-Cores, Fuzzy Clustering—Fuzzy C-means, RockCluster, Biclust, Partitioning Around Medoids (PAM), CLUES,Self-Organizing Maps (SOM), Proximus,CLARA
- **Classification**—SVM, penalizedSVM,kNN, Outliers, Decision Trees, Naive Bayes, adaboost, JRip
- **R Packages**—RWeka, gausspred, optimsimplex, CCMtools, FactoMineR, nnet

Machine Learning is defined as the science of getting computers to act without being explicitly programmed A complete list is defined at http://en.wikipedia.org/wiki/List_of_machine_learning_algorithms.

It has the following constituents:

- **Supervised learning**—Statistical classification
- **Unsupervised learning**—Artificial neural network, Association rule learning, Hierarchical clustering, Cluster analysis, Outlier Detection
- **Reinforcement learning**
- **Deep learning**

A nice book for understanding machine learning in R is Machine Learning in R (Packt Publishing). The Machine Learning View in R (http://cran.r-project.org/web/views/MachineLearning.html) has a great collection of all R packages dealing with Machine Learning.

In R, rattle is a package and GUI that allows the user to use a lot of algorithms with comparative ease. It has cluster analysis, regression models, classification models, SVM, neural nets, random forests, decision trees, and association analysis, all done using a few clicks and with very good documentation and lucid examples.

2.13 Interview of John Myles White, co-author of Machine Learning for Hackers

A partial extract from an interview of John Myles White, co-author of Machine Learning for Hackers.

Ajay—How can academia and private sector solve the shortage of trained data scientists (assuming there is one)?

John—There is definitely a shortage of trained data scientists: most companies are finding it difficult to hire someone with the real chops needed to do useful work with Big Data. The skill set required to be useful at a company like Facebook or Twitter is much more advanced than many people realize, so I think it will be some time until there are undergraduates coming out with the right stuff. But there is a huge demand, so I am sure the market will clear sooner or later.

The changes that are required in academia to prepare students for this kind of work are pretty numerous, but the most obvious required change is that quantitative people need to be learning how to program properly, which is rare in academia, even in many CS departments. Writing one-off programs that no one will ever have to reuse and that only work on toy data sets does not prepare you for working with huge amounts of messy data that exhibit shifting patterns. If you need to learn how to program seriously before you can do useful work, you are not very valuable to companies who need employees that can hit the ground running. The companies that have done best in building up data teams, like LinkedIn, have learned to train people as they come in since the proper training is not typically available outside those companies. Of course, on the flipside, the people who do know how to program well need to start learning more about theory and need to start to have a better grasp of basic mathematical models like linear and logistic regressions. Lots of CS students seem not to enjoy their theory classes, but theory really does prepare you for thinking about what you can learn from data. You may not use automata theory if you work at Foursquare, but you will need to be able to reason carefully and analytically. Doing math is just like lifting weights: if you are not good at it right now, you just need to dig in and get yourself in shape.

About—John Myles White is a Phd Student in Ph.D. student in the Princeton Psychology Department, where he studies human decision-making both theoretically and experimentally. Along with the political scientist Drew Conway, he is the author of a book published by OReilly Media entitled Machine Learning for Hackers, which is meant to introduce experienced programmers to the machine learning toolkit.

http://www.springer.com/978-1-4939-1701-3

R for Cloud Computing
An Approach for Data Scientists
Ohri, A.
2014, XVII, 267 p. 255 illus., 160 illus. in color.,
Softcover
ISBN: 978-1-4939-1701-3