COURSE BOOK

Data Science

DLMBDSA01

iubh INTERNATIONALE HOCHSCHULE

Course Book

Data Science

DLMBDSA01

iubh INTERNATIONALE HOCHSCHULE

Masthead

Publisher:
IU Internationale Hochschule GmbH
IU International University of Applied Sciences
Juri-Gagarin-Ring 152
D-99084 Erfurt

Mailing address:
Albert-Proeller-Straße 15-19
D-86675 Buchdorf

media@iu.org
www.iu.de

DLMBDSA01
Version No.: 001-2021-1213

Module Director

Prof. Dr. Ulrich Kerzel

Mr. Kerzel teaches data science and artificial intelligence with a focus on the development and economic/industrial applications of modern methods of machine learning. After obtaining his doctorate, he spent several years conducting research at the University of Cambridge and the European Organization for Nuclear Research (CERN) before becoming a senior data scientist at the software company Blue Yonder. There, Mr. Kerzel led the development team for machine learning, supervised practical projects in industry and commerce, and prepared a wide range of companies for their transition to a data-driven enterprise.

Table of Contents

Data Science

Module Director 3

Introduction
Data Science 7

Signposts Throughout the Course Book 8

Learning Objectives 9

Unit 1
Introduction to Data Science 12

1.1 Overview of Data Science 13

1.2 Data Science Activities 18

1.3 Sources of Data 20

1.4 Descriptive Statistics 23

Unit 2
Use Cases and Performance Evaluation 34

2.1 Data Science Use Cases (DSUCs) 34

2.2 Performance Evaluation 40

Unit 3
Data Preprocessing 52

3.1 Transmission of Data 52

3.2 Data Quality, Cleansing, and Transformation 54

3.3 Data Visualization 57

Unit 4
Processing of Data 68

4.1 Stages of Data Processing 69
4.2 Methods and Types of Data Processing 73
4.3 Output Formats of Processed Data 74

Unit 5
Selected Mathematical Techniques 82

5.1 Principal Component Analysis 82
5.2 Cluster Analysis 91
5.3 Linear Regression 99
5.4 Time-Series Forecasting 105
5.5 Transformation Approaches 114

Unit 6
Selected Artificial Intelligence Techniques 120

6.1 Support Vector Machines 120
6.2 Artificial Neural Networks 122
6.3 Further Approaches 140

Appendix 1
List of References 146

Appendix 2
List of Tables and Figures 150

Introduction

Data Science

iubh INTERNATIONALE HOCHSCHULE

Signposts Throughout the Course Book

Welcome

This course book contains the core content for this course. Additional learning materials can be found on the learning platform, but this course book should form the basis for your learning.

The content of this course book is divided into units, which are divided further into sections. Each section contains only one new key concept to allow you to quickly and efficiently add new learning material to your existing knowledge.

At the end of each section of the Interactive Book, you will find self-check questions. These questions are designed to help you check whether you have understood the concepts in each section.

For all modules with a final exam, you must complete the knowledge tests on the learning platform. You will pass the knowledge test for each unit when you answer at least 80% of the questions correctly.

When you have passed the knowledge tests for all the units, the course is considered finished and you will be able to register for the final assessment. Please ensure that you complete the evaluation prior to registering for the assessment.

Good luck!

Learning Objectives

This course book in **Data Science** provides the framework to create values from data through the study of data science. After an introduction on the importance of this field, you will learn how to outline the main activities in data science and label the different data sources. Because the foundation of data science—predictive analytics—is the understanding of underlying data characteristics, we will go through the concepts of descriptive analytics and probability theory, with a focus on Bayesian statistics.

Afterwards, you will learn how to identify a data science use case in diverse organizations and predict the value of each use case. Furthermore, you will be able to analyze a developed prediction model through evaluation metrics, measuring how it will be effectively implemented in the organization's business through key performance indicators.

Because raw data come from several sources and in all different shapes and types, you will learn how to apply preprocessing techniques to raw data in order to improve its quality and validity for the forthcoming predictive analysis. An emphasis is given to data visualization tools which help with understanding the input data.

Consequently, you will learn how to develop a data prediction model through common mathematical techniques of machine learning and artificial intelligence approaches. You will also learn how businesses take a model's outputs and use them to make better decisions and take more appropriate actions.

Unit 1

Introduction to Data Science

STUDY GOALS

On completion of this unit, you will have learned ...

... the meaning of data science.

... common terms and definitions in data science.

... the different applications of data science.

... the typical sources of data.

... the types and shapes of data.

... probability distributions and Bayesian statistics.

1. Introduction to Data Science

Introduction

Data
These are facts, observations, assumptions, or incidences.

Information
Patterns and relationships among data elements are instances of information.

Data science
The field of data science concerns the systematic practice of analyzing data, exploring the information contained in the data, and creating useful predictions to advise and guide the decision-making process.

A checkout counter at a supermarket issues receipts that include data about products' identification numbers and the cost of each item. These **data** can be analyzed to get meaningful information, e.g., the total number of milk bottles sold or which brand(s) sold most rapidly. Hence, the supermarket's manager can have a deep understanding of all the parameters of the supermarket's operations (replenishment, promotion planning, etc.) and transactions, estimate the hidden variables which control the processes, and predict what the supermarket can expect over the coming days and weeks. This type of analysis—which leads to the extraction of useful and/or hidden **information**, actionable insights, and reasonably accurate predictions—is an important aspect of **data science**.

By applying data science elements, a business collects numerous data on its operations which are presented in easily accessible and understandable forms for managers to use. In the case of a supermarket, managers obtain solid knowledge of costs and revenues and expectations for costs/revenues if a different business scenario is implemented. In another example, Das and Sharma (2016) reported that through the application of data science tools, the United Parcel Service (UPS) "installed sensors to collect data on speed and location of its vans, which combined with GPS information, reduced fuel usage in 2011 by 8.4 million gallons, and shaved 85 million miles off its routes" (Das, 2017, p.1).

A further example of strategic use of data science tools is the Internet Movie Database (IMDB). IMDB provides data about all elements of the movie industry online. Every actor who has played a role in a movie listed in IMDB's database has their own entry on the website.

IMDB is implementing a data science tool to extract relevant information and answer questions such as:

- Which actors appeared in the highest-rated movies?
- Which actors are predicted to play starring roles in the coming year?
- Which movie is expected to be the highest-rated film this year?
- How closely do viewer ratings of movies correlate to movie awards?
- What is the average age difference between actors playing husband and wife in a movie? Has this age difference increased or decreased over the last ten years? What is the forecasting for this difference over the next few years?

1.1 Overview of Data Science

Data in all forms (e.g., numbers, words, statistics, and measurements) are important units for businesses. With advances in computing resources and associated technologies, large amounts of data are generated and collected every day by different entities.

Examples abound, including monetary transactions through a financial institution, patterns of seismic activity in a region (sensor-generated data), and internet surfing history.

But all this data is not very useful in its raw form. Data science is concerned with the arrangement, analysis, visualization, and interpretation of collected data for the purpose of extracting embedded knowledge and useful information and predicting scenarios should new or different data be introduced. As a result, the output provided by applying data science can aid decision-makers in choosing appropriate actions, reviewing applicable thoughts, and thinking about optimum scenarios for their businesses. In some cases, applying data science may lead to automation of the entire business process, reducing the need for human intervention.

The main drivers of the emerging need for data science are

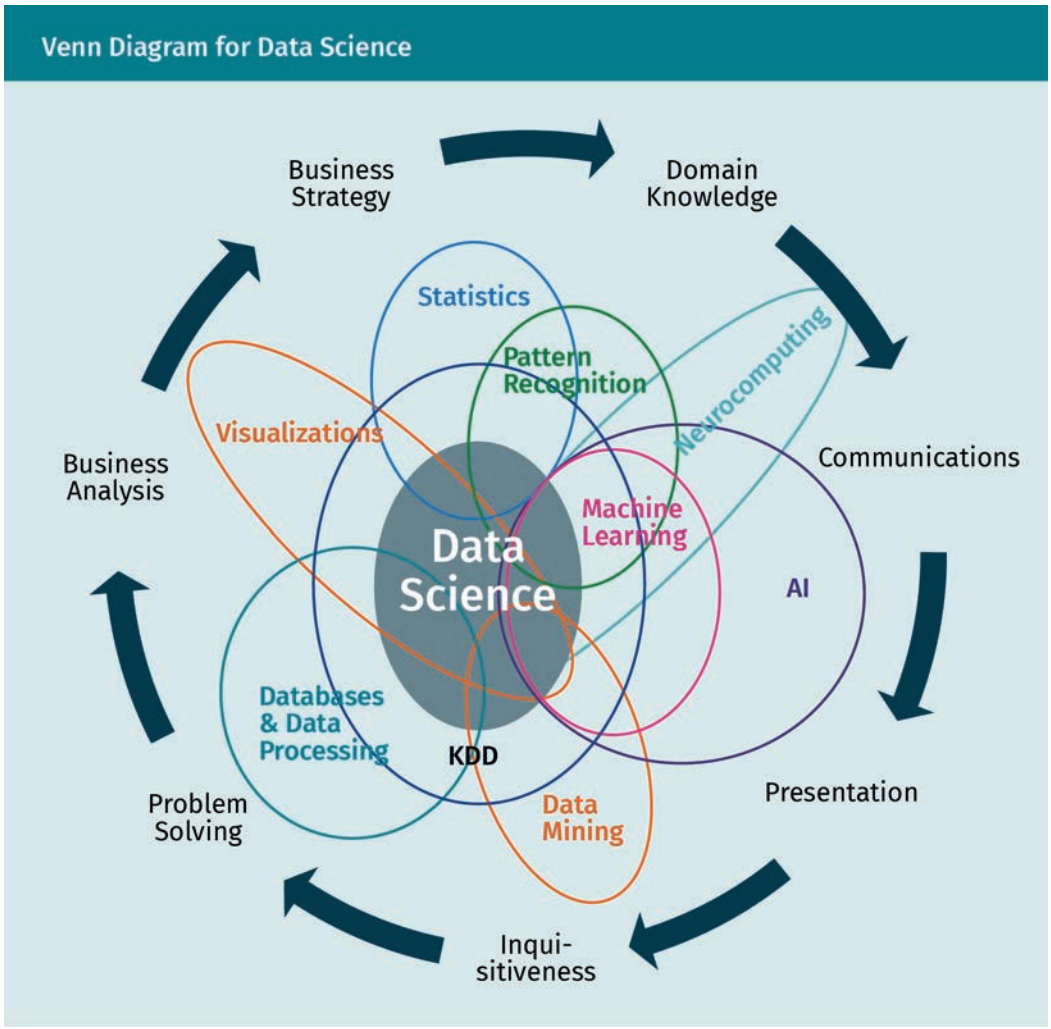
- advances in computer capabilities and computing platforms;
- availability of data from various sources; and
- ability to store large amounts of data.

Data science is an intrinsically data-driven and cross-/interdisciplinary field comprising many distinct overlapping capabilities. According to Baldassarre (2016), these capabilities include

- artificial intelligence (machine learning approaches in particular);
- **data mining**;
- pattern recognition;
- knowledge discovery (KDD);
- database storage and data processing;
- data visualization;
- statistics; and
- neurocomputing.

Data mining
This is the process of discovering patterns in large datasets.

These capabilities cooperate to create platforms for generating and sharing data, apply numerical techniques to organize and model data, and extract useful information from collected data, as illustrated by the data science Venn diagram.



One frequently-used, but confusing, expression in the business management field is **business intelligence**, which should not be confused with data science. The aim of business intelligence (BI) is to generate a descriptive analysis of data for use in reporting the past behavior of a business; the aim of data science—in this case—is to use predictive analysis for estimating a business's future behavior.

Business intelligence
This is a set of strategies for identifying, extracting, analyzing, managing, and delivering important trends relevant to business metrics.

Data Science Terms

What follows is a collection of common terms in data science.

Artificial intelligence
Artificial intelligence (AI) is concerned with the development of models that enable a computer to "think" and learn by itself through the feeding of relevant data. There are two types of AI: narrow and general. According to Hackernoon (2018), "narrow AI refers to AI which is able to handle just one particular task. General AI is something altogether more sophisticated. This refers to a system which is able to cope with any generalized task which is asked of it, much like a human."

Machine learning

Machine learning is a subset of artificial intelligence in which numerical models are developed in order to predict the probability of a future event or new data record. The prediction result is expressed in the form of a probability between 0 and 100 percent, a probability density function, or a number with associated uncertainty. Results are evaluated using predefined accuracy metrics and then transformed into a business decision using a given objective and key performance indicators.

Supervised learning
This is a machine learning task of inferring a numerical function for data with a predefined target output.

Unsupervised learning
This is a blind machine learning task of inferring a binary function (on/off) where the target output is not defined.

There are two main types of machine learning: **supervised** and **unsupervised**. Cluster analysis is a type of unsupervised learning that sorts a set of data records into groups (clusters). The records in a cluster are more similar to one another than to those in other clusters. The level of similarity is defined by a domain expert.

Supervised learning can be one of two types: classification or regression. Classification is a forecasting technique which categorizes a dataset into predefined classes. Regression is a forecasting technique for determining how a target variable is related to the input variables and then predicting a value for this target variable given new data scenarios.

There is a third type of machine learning, called reinforcement learning. An objective is defined, and the computer learns the best approach to achieve this objective.

Training and testing sets

The training set is comprised of the particular records in a dataset that are used by the machine learning model to build a predictive mathematical function. The testing set is made up of different records in a dataset that are used to measure the performance of the machine learning model that has been developed.

Data variables

The measurable and observable quantities about data records are the data variables (or data features). An example would be the height, length, and breadth of a solid object. It may be necessary to apply a kind of feature engineering to the data so that new, relevant features are constructed to convey the information optimally.

The process of reducing a dataset to a list of selected variables while ensuring that it conveys similar information is called dimensionality reduction. The list of selected variables excludes those variables which are redundant and/or have little influence on the changeability of the dataset. The objective of applying dimensionality reduction is to simplify the computational cost required to develop the machine learning model.

Types of errors

The output of the classification prediction model is measured according to parameters such as type I error, type II error, sensitivity, and specificity. Type I error represents the number of false positives, i.e., the number of negative events that were predicted to be positive by the model. For example, if a model predicts real cases of fraud as being not fraud, this is a type I error. A type II error represents the number of false negatives, i.e., the number of positive events predicted to be negative by the model. A type II error occurs if a model predicts a certain case to involve fraud when it actually does not.

Sensitivity and specificity represent the true positive rate and true negative rate, respectively.

$$\text{sensitivity} = 1 - \frac{\text{type II error}}{\text{number of real positives}}$$

$$\text{specificity} = 1 - \frac{\text{type I error}}{\text{number of real negatives}}$$

For the regression prediction model, output is measured according to metrics such as absolute error, mean square error, and relative error.

Data Science Applications

Data science produces predictions about data patterns that are valid, useful, and comprehendable. Therefore, since data are considered the main element in many applications, there is significant attention paid to data science techniques.

Industrial processes applications
Data are obtained at different levels of the production process, e.g., ingredient and actuator data at the field level; signal data at the control level; monitoring sensor data at the execution level; and indicator data at the planning level. The main goals of applying data science to industrial processes are the automation and optimization of these processes and simultaneous improvement in the company's competitive standing.

Business applications
Businesses can be sources of data about aspects such as customers, portfolios, human resources, marketing, sales, and pricing. The main goal of applying data science to business data is to better understand, motivate, and drive business processes. For example, bottlenecks in business processes can be identified and predictions about sales can be estimated.

Text data applications
Data contained within text serve in many applications as an important information resource. Examples are text documents, e-mails, and Web documents. The main goal of applying data science to text data is to filter, search, extract, and structure information.

Image data applications
Data formatted as images are easy to obtain due to advances in imaging sensor technology. Sensors range from cameras in smartphones to satellite cameras and provide large amounts of two- and three-dimensional image data. The main goal of applying data science to image data is to find and recognize objects, analyze and classify scenes, and relate the images to other information sources.

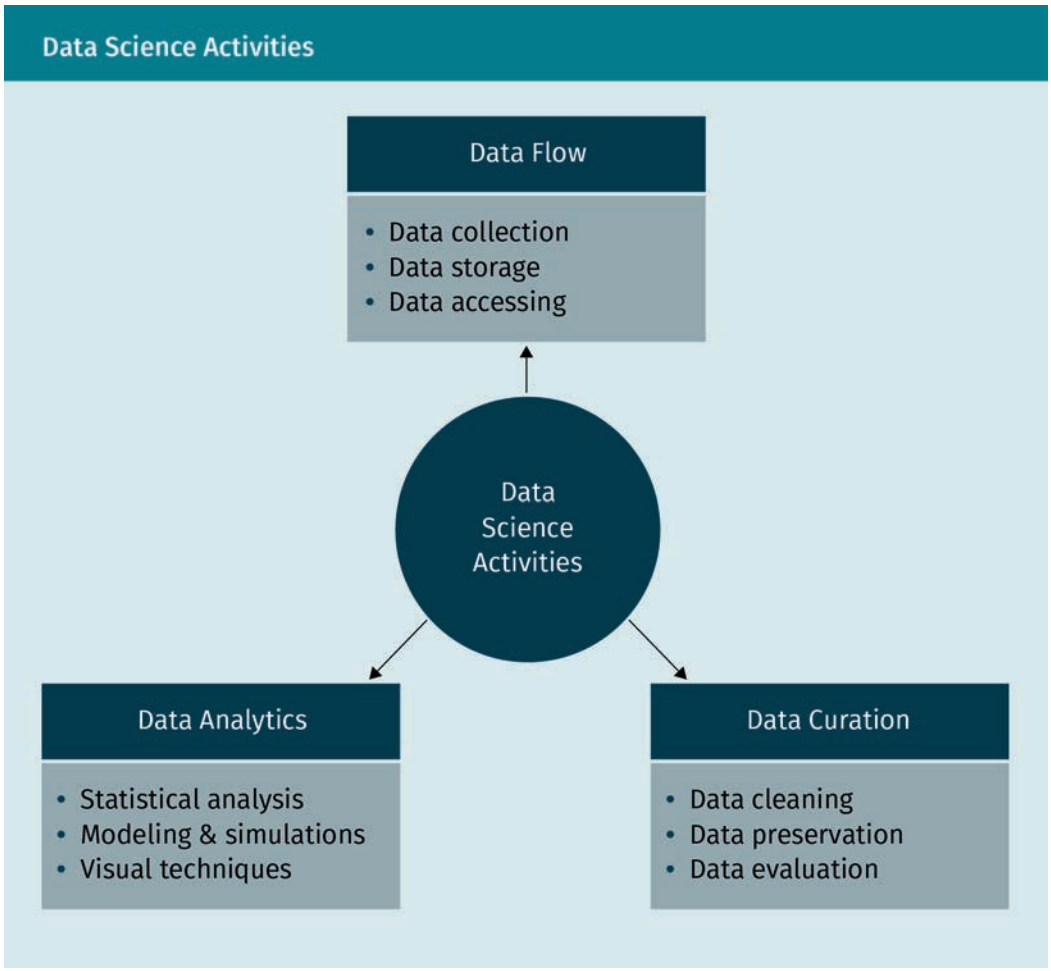
Medical data applications
Medical data are collected from patient care (e.g., patient health records), clinical care programs, and medical laboratory experiments, among others. A few of the reasons to apply data science to medical data are to analyze, understand, and annotate the influence/side effects of medicines and detect and predict different levels of certain diseases.

Case studies demonstrating how data science can be utilized for some of the applications just mentioned abound.

- Marketing and sales teams benefit from the availability of customer data. For example, data science is implemented in order to identify which marketing channels (e-mail, telephone, television, etc.) are the most effective.
- Human resources departments can apply data science to employee data to detect which competences (skills, education, etc.) have the most impact on employees' capabilities to improve their performance.
- Customer service departments use customer comments about a certain product to ascertain customer satisfaction and identify aspects that need to be considered by management for improvement.
- Smart cities rely on infrastructure sensors for collecting huge amounts of data, and data science tools are applied to transform data points into actions that improve residents' lives. For example, improvements to the public transportation system might include more bus stops, shorter waiting periods between trains, reduction in cost per ride, etc.
- In medical applications, recent advances in artificial intelligence permit the diagnosis of diseases when no specialist is available.

1.2 Data Science Activities

Data science activities are conducted in three dimensions: data flow, data curation, and data analytics. Each dimension concerns a group of data science challenges and associated solution methodologies and numerical techniques.



First Dimension: Data Flow

Managing data flow starts with the collection of data, including a list of possible sources and attributes. The storage structure of the data is designed to coincide with the format of the collected data, and the stored data must be transparent, complete (to an extent), and accessible.

Second Dimension: Data Curation

Data curation is the process of refining collected data, and there are different ways of doing so.

1. "Data preservation" means the data are cleaned from noise (e.g., typing errors in data entries) and fake outliers, and if there are missing values, one has to determine how they should be handled. Improving data quality typically requires detailed knowledge of the domain in which the data are recorded.
2. "Data description" includes data structure, schemes, and metadata.

3. “Data publication” helps make data available so it can be used effectively.
4. “Data security” is necessary to secure and protect data, as well as determine the legal frameworks and policies to be followed. Physical threats and human errors should be identified wherever possible.

Third Dimension: Data Analytics

Data analytics uncovers hidden patterns in the data and transforms data into relevant, useful information. It predicts future events in order to support and/or automate decision-making processes. Analytics techniques include modeling and simulation, machine learning, artificial intelligence, and statistical analysis.

1.3 Sources of Data

Data sources should be trustworthy to ensure the collected data are robust and high quality. Common sources of data are companies, governments, academic institutions, websites, and media platforms.

Organizational and trademarked data sources
Large companies like Google and Facebook possess enormous amounts of data. They provide bulk downloads of public data for offline analysis in order to enrich the organization’s market visibility. Google and Facebook also have internal data for use by their employees. Almost all companies possess their own data collected from internal systems that record various activites. The first and most important users of this proprietary data are the company’s employees, although some companies sell datasets (e.g., weather forecasts).

Government data sources
Federal governments are committed to providing open data to enable and enhance how the government fulfills its missions. Governmental organizations also release demographic and economic data (e.g., population per geographic region) every few years to be analyzed for the sake of more accurate risk estimation.

Academic data sources
Academic research creates large datasets, and many scientific journals require that these datasets be made available to other researchers. These datasets span many fields including medicine, economics, and history.

Web page data sources
Web pages often provide valuable numerical and textual data. For example, you can request all tweets containing a certain hash tag from Twitter (e.g., iPhone X) and apply sentiment analysis to determine whether a tweet has a positive or negative slant on that topic (i.e., hash tag). The customer support division of an organization associated with this topic (e.g., Apple) can use this information to improve their business operations.

Media data sources
Media includes videos, audios, and podcasts that provide quantitative and qualitative information on the characteristics of user interaction. Since media crosses all demographical borders, it is the quickest method for businesses to identify and extract patterns in data to enhance decision-making.

Data Types

Think about any data you are keen to collect—perhaps data about human characteristics—and propose possible relevant attributes (eye color, gender, IQ score, height, weight, etc.). These attributes can be described using quantitative and qualitative data types.

Quantitative data
Quantitative data are measurable values. Subtypes include

- categorical nominal. Categories of this subtype do not have an inherent order (e.g., eye color).
- categorical ordinal. Categories of this subtype have inherent order (e.g., salary grade, age group). These data come either in ordered or unordered sequences. For example, {A, B, C} does not indicate that "A" has the highest/lowest priority because such a ranking should be according to another predefined parameter.
- categorical binary. In this subtype, data are divided into one of two categories (e.g., gender (male/female), power button status (on/off)).
- discrete. In this subtype, the data attribute is usually any digit in the numbering system (e.g., number of students, student ages).
- continuous. In this subtype, a data attribute is a value within a range (e.g., temperature, student exam results).

Qualitative data
Qualitative data provide information about the quality of a good or service. Examples of qualitative data are customer feedback about a product, focus group discussions, and answers to open-ended questions.

Data Shapes

Data come in one of three shapes (structured, unstructured, or semi-structured) or as a stream. **Structured data** are well-defined, and all the attributes of the data (i.e., columns) and its records (i.e., rows) are known. **Unstructured data** include text and multimedia content. An e-mail message is one example. Unstructured data are data in their raw shape. For example, one part of a dataset might be provided as pictures, while another part is provided as text or sound. Before processing, these data files must be transformed into a structured format by applying complex numerical tools that find common attributes. **Semi-structured data** are not necessarily in complete tabular form—some of the data may be in different form(s). However, these data do have organizational properties that make them easier to analyze, and with some degree of process-

Structured data
These data have a high level of organization, such as information shaped in tabular forms of rows and columns.

Unstructured data
These are data with unknown form or structure.

Semi-structured data
These are all data shapes between structured and unstructured.

ing, they can be converted into structured data. Streaming data are continuously generated by different sources and at high speeds. Such data are processed incrementally and allow users immediate access, meaning users do not have to wait for the data to download.

The Five Vs of Data

The main obstacles to handling any type of data and describing data overloads are volume, variety, and velocity (the original "3 Vs of data") and often also veracity/validity and value.

Volume

Volume
The amount and scale of data is their volume.

Today there are data of enormous **volumes**. Audio and video formats have volumes in the range of terabytes and zettabytes. An airplane fitted with 5,000 sensors will generate 10 GB of data for every second it is in flight (Rapolu, 2016). It is expected that the amount of data in the world will double every two years. However, with advances in computational power and decreasing storage costs, the creation of so much data is not expected to confront any challenges.

Variety

Variety
Data come from many sources, are of many types, and have different levels of complexity.

There is a considerable **variety** of data. Previously, most data were created in a structured shape to simplify data science tasks; today much of the data generated by organizations can be considered unstructured.

Velocity

Velocity
The speed at which data are created, stored, analyzed, and visualized is the velocity.

Velocity of data refers to how fast they are created and how fast they must be processed. Over 500 hundred hours of videos are uploaded onto YouTube every minute (Statista, 2019). Computational tools need considerable time to process data and update databases. With the availability of high-performance computing, data can be analyzed and passed on to the database in a very short interval of time.

Veracity/Validity

Veracity
Data's quality is their veracity.

Validity
This refers to the value of data in extracting useful information for making a decision.

In general, data contains **veracity** with elevated levels of noise obtained during data collection or through the data processing phase. Noise influences the degree to which the data can be trusted. Hence it is essential to use cleaning tools during the pre-processing stage. Data may be correct and noise-free but outdated and therefore no longer **valid**. Meaningful conclusions cannot be deduced from invalid data.

Value

Value of data refers to which value can be obtained by building a data-driven application around it, in particular in a business setting. Tapping into hitherto unused sources of data allows companies to build new services, improve their customer service or customer experience, improve operational processes, etc.

1.4 Descriptive Statistics

Value
Data's value (often called its usage value) refers to the application they are used for and the frequency of their use.

For each variable of a given dataset, probability theory determines a set of statistical parameters which report the dataset characteristics as a whole batch. Descriptive analysis is then performed to provide end users with an understandable description of a specific dataset's characteristics in various forms. Several statistical parameters are calculated to describe the variable clearly (Skiena, 2017). These statistical parameters include a variable's minimum and maximum values, mean, median, and **standard deviation**.

Standard deviation
This is a measure for how spread apart data values are. It is typically applied for normally-distributed data.

The mean is the arithmetic average of the variable's values, while the median is the value located exactly at the middle point of a variable's sorted values. The mean is more sensitive to extreme values than the median. For example, if the variable's values are (2, 3, 4, 5, 6, 7, and 9), then:

$$\text{Mean} = \frac{2+3+4+5+6+7+9}{7} = 5\,.\,14$$

$$\text{Median} = 5$$

Probability Theory

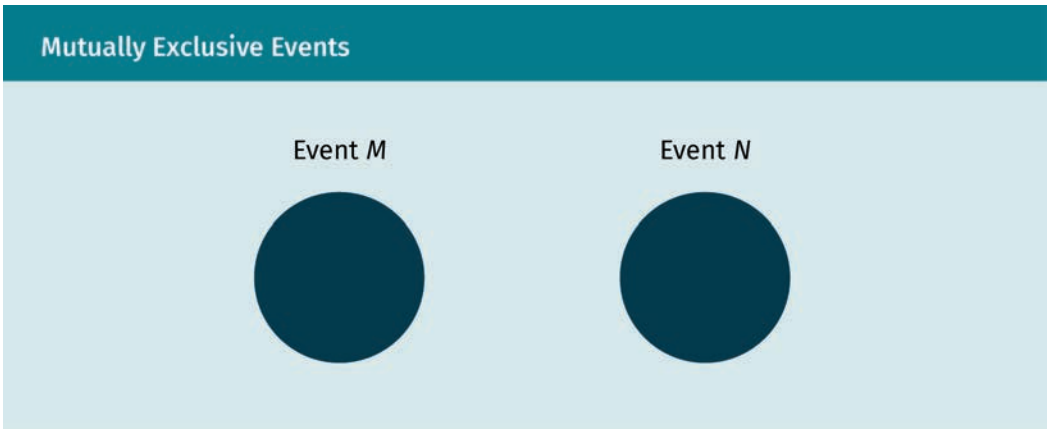
Many important techniques in data science are based on **probability** theory. If an event is unlikely to happen, its probability is "$\mathrm{P} = 0$"; if an event is certain to happen, its probability is "$\mathrm{P} = 1$". The probability of a random event is always between "0" and "1", depending on the chance of its occurrence. The probability cannot be a negative value, and the sum of all probabilities must always be "1".

Probability (P)
The chance of an event happening is its probability (P).

If we are making profit (Event M) through one client account, we cannot make a loss (Event N) through that same account. In this case, a particular event (loss) cannot occur at the same time as another event (profit). These two opposite events are known as mutually exclusive events.

$$\mathrm{P}(\mathrm{M} \text{ and } \mathrm{N}) = \mathrm{P}(\mathrm{M} \cap \mathrm{N}) = 0$$

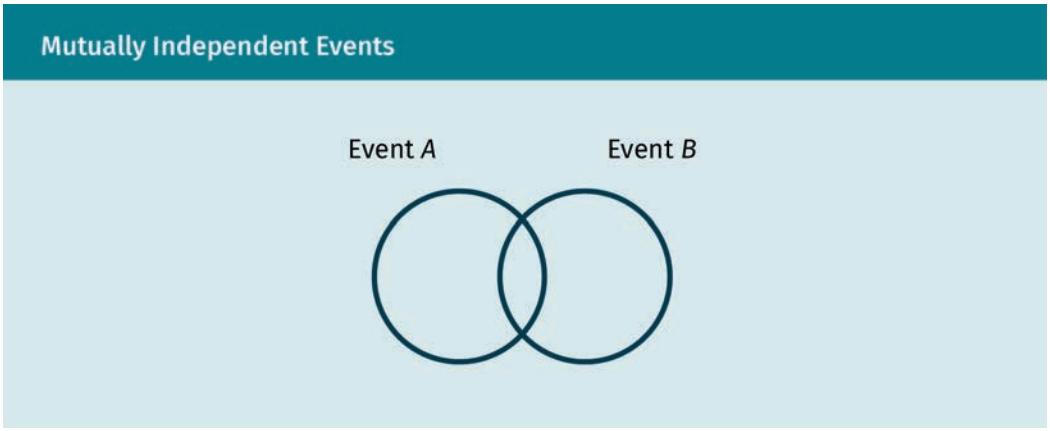
$$\mathrm{P}(\mathrm{M} \text{ or } \mathrm{N}) = \mathrm{P}(\mathrm{M} \cup \mathrm{N}) = \mathrm{P}(\mathrm{M}) + \mathrm{P}(\mathrm{N})$$



In another case, a company may be making a profit (Event A) but at the same time having legal issues (Event B). Both events can occur simultaneously and do not impact one another. They are said to be mutually independent events.

$$\mathrm{P(A \text{ and } B) = P(A \cap B) = P(A) \cdot P(B)}$$

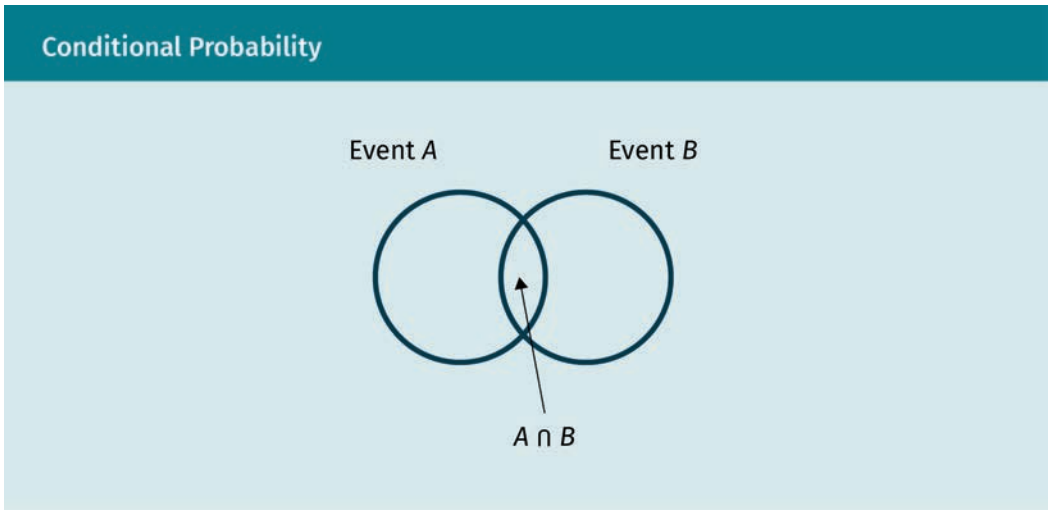
$$\mathrm{P(A \text{ or } B) = P(A \cup B) = P(A) + P(B) - P(A \cap B)}$$



Conditional probability

When two events are correlated, the conditional probability $\mathrm{P(A|B)}$ is defined as the probability of an event A, given that event B has already occurred.

$$\mathrm{P(A|B) = \frac{P(A \cap B)}{P(B)}}$$



In data science, all predictions from developed models are probabilities, having either a probability between 0 and 1 (for classification models) or a probability density distribution (for regression models). Operationally, model predictions need to be presented as one number which will be used by management to determine future actions and decisions. In probability theory, a given value for a particular variable is considered a random event, and we calculate its probability of occurrence (i.e., how often this value appears within the variable's values). We consider each variable's value a random event and then calculate its probability. This forms the probability density function (pdf).

Probability density function

The probability density function is a graph where the x-axis represents the range of a variable's possible values and the y-axis denotes the probability of each value. For example, a given dataset is for the tossing of two dice. One of the variables is for the sum of the outcomes of the two dices in each toss. This variable will have a minimum value of 2 (when each of the two dice shows "1" as its output) and a maximum value of 12 (when each of the two dice shows "6" as its output). For each possible output (i.e., event) of the first die (e.g., "4"), there are six possible outputs for the second die ("1", "2", "3", "4", "5", or "6"). As a result, the total number of possible events are $(6\ [\text{first die events}] \cdot 6\ [\text{second die events}] = 36\ \text{events})$.

The probability of having a variable value of 5 (i.e., the sum of the two dice is 5) is determined by obtaining these possible events:

$$(\text{"first die"}, \text{"second die"}) = (\{\text{"1"}, \text{"4"}\}, \{\text{"2"}, \text{"3"}\}, \{\text{"3"}, \text{"2"}\}, \{\text{"4"}, \text{"1"}\}) = 4\ \text{events}$$

This means the probability of having a variable value of 5 is:

$$\mathrm{P}(5) = \frac{4\ \text{possible events}}{36\ \text{events}} = 0.11$$

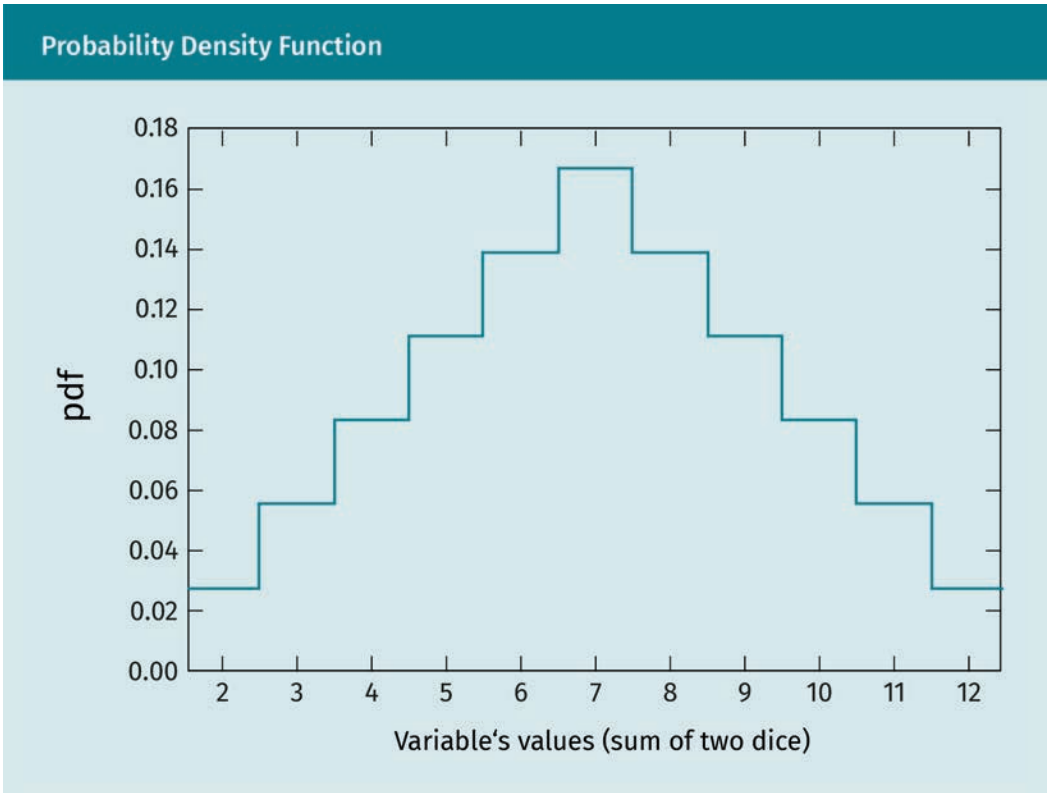
To get variable value of 6, the possible events are:

$$(\text{“first die”, “second die”}) = (\{\text{“1”, “5”}\}, \{\text{“2”, “4”}\}, \{\text{“3”, “3”}\}, \{\text{“4”, “2”}\}, \{\text{“5”, “1”}\}) = 5 \text{ events}$$

This results in the probability:

$$\mathrm{P}(6) = \frac{5 \text{ possible events}}{36 \text{ events}} = 0.138$$

In the same manner, we can calculate the probability of obtaining every possible variable value to form the probability density function.



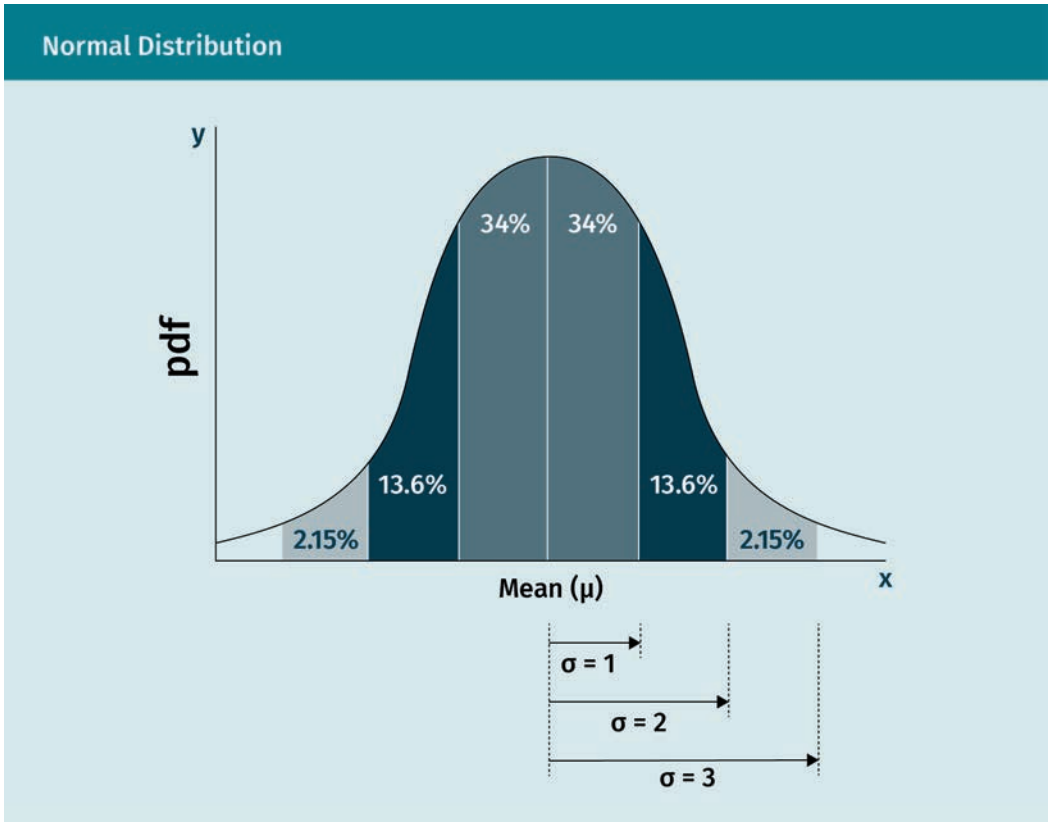
Probability Distributions

Every variable of a dataset meets a particular frequency distribution (i.e., definite probability density function) which reflects how often each value of this specific variable occurs. Although the shapes of these frequency distributions are not unique, there are some general and classical distributions which regularly appear for a wide range of datasets. A pleasant property of these distributions is that they can be mathematically described with closed form expressions of a few parameters. The most important probability distributions are reviewed in this section.

Normal distribution

The normal distribution has a bell-curve shape and represents a significant amount of real-life data attributes, because most attribute values center on their mean value. An example of normal distribution is the distribution of employee performance in an organization, where only a few employees are identified as high or low performers; most employees perform somewhere in the middle (i.e., in close proximity to the mean).

The normal distribution has 68% of values within standard deviation $\sigma = \pm 1$; 95% of values within standard deviation $\sigma = \pm 2$; and 99.7% of the values within standard deviation $\sigma = \pm 3$.



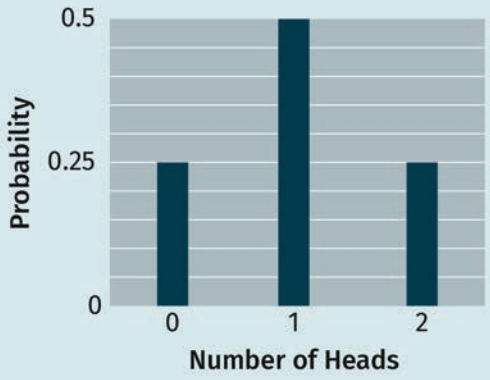
Binomial distribution

The binomial distribution monitors the success of an event occurring. An example of binomial distribution is a coin toss in which you win if "heads" is the outcome and lose if the outcome is "tails." If you toss the coin twice, what is the probability of having "heads" only once? What is the probability of having two "heads"? And what is the probability of not having "heads" at all? The table below presents all the possible outcomes when the coin is tossed twice.

Possible Outcomes of a Coin Toss		
Outcome	First toss	Second toss
1	Heads	Heads
2	Heads	Tails
3	Tails	Heads
4	Tails	Tails

Out of the four possible outcomes, having "heads" twice can happen only once $\left(\mathrm{P}(\text{two heads}) = \frac{1}{4} = 0.25\right)$. Having "heads" for one of the two tosses can happen twice $\left(\mathrm{P}(\text{one head}) = \frac{2}{4} = 0.5\right)$. And not having a "heads" for either of the two tosses can happen only once $\left(\mathrm{P}(\text{no heads}) = \frac{1}{4} = 0.25\right)$. The probabilities of these possibilities are represented by binomial distribution.

Binomial Distribution



Poisson distribution

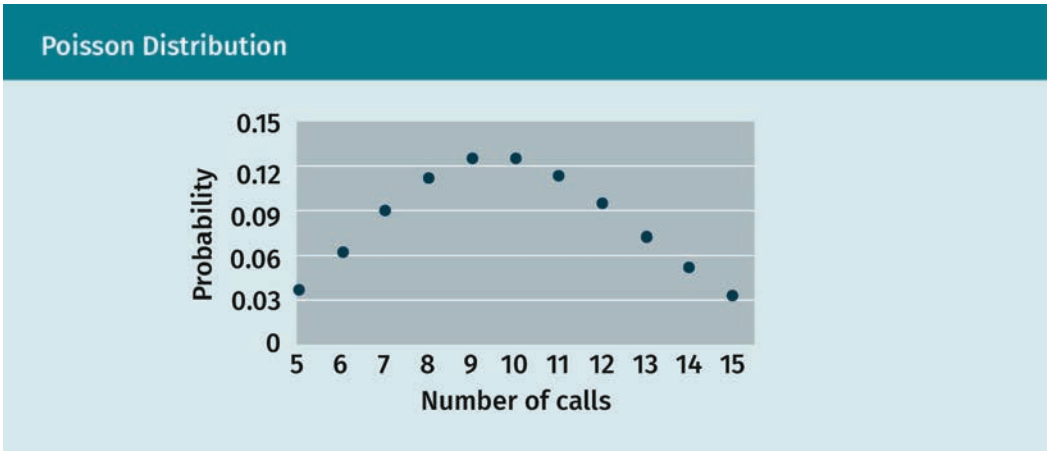
The Poisson distribution monitors the frequency of intervals between independent events. It calculates the probability of a certain number of event occurrences based on the mean number of occurrences. Mathematically, the Poisson distribution is represented by the following equation:

$$\mathrm{P}(\mathrm{x}) = \frac{\mathrm{e}^{-\mu}\mu^{\mathrm{x}}}{\mathrm{x}!}$$

where μ is the mean number of occurrences, and x is the required number of occurrences. For example, if a call center receives an average of ten calls per day, what is the probability that in a given day the call center receives exactly seven calls?

$$P(7) = \frac{e^{-10}10^7}{7!} = 0.09$$

In the same manner, the probability can be calculated for a various number of calls, forming the Poisson distribution.



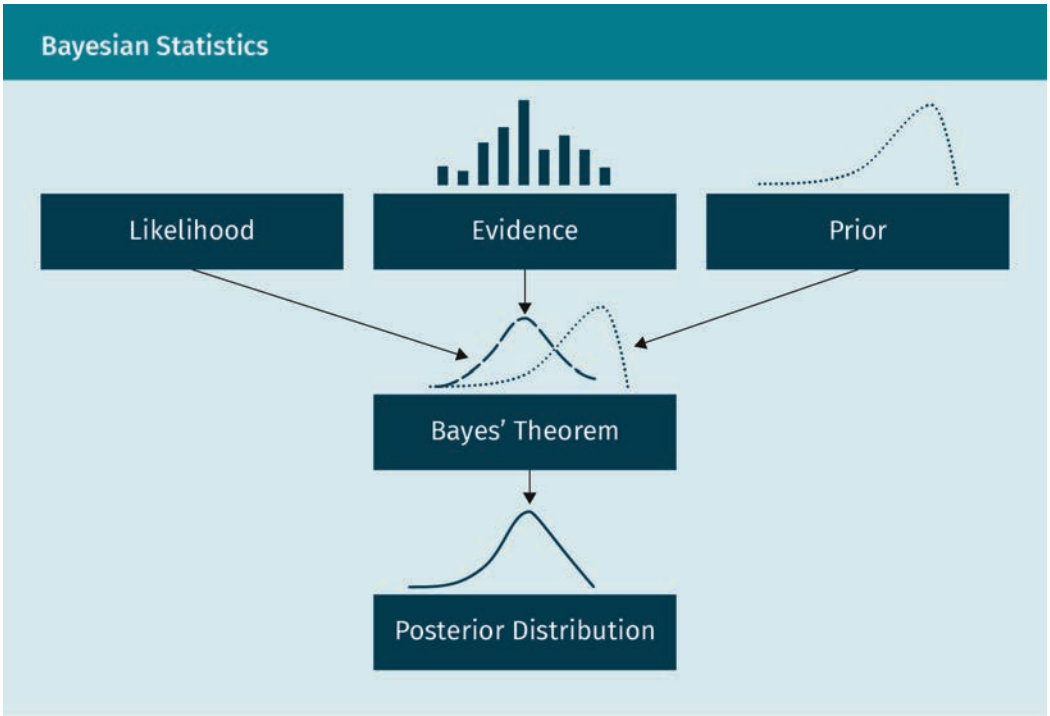
There are many processes which are considered Poisson processes, such as sales records, cosmic rays, and radioactive decay, because they are independent of each other and only one parameter is required (e.g., mean number of occurrences per time).

Bayesian Statistics

In general, the Bayes theorem is formulated using the following conditional probability equation for random events A and B:

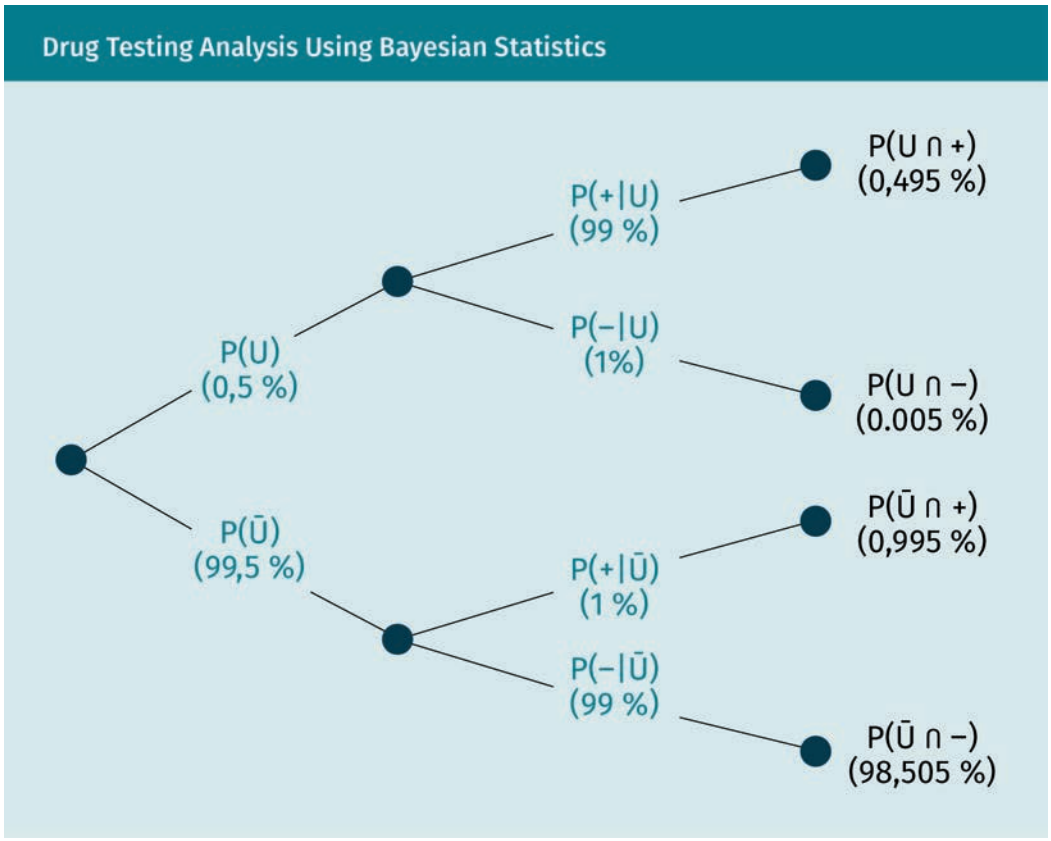
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Here, P(A) is called the "prior" and represents the strength of our belief in the occurrence of event A. This probability can have any value between 0 and 1. The likelihood, $P(B|A)$, represents the probability of observing B, given the occurrence of A. The evidence, P(B), represents the probability of the occurrence of all possible values of B, weighted by how strongly we believe in those particular values of B. Finally, $P(A|B)$ is the "posterior belief" of variable A after observing the evidence B.



The Bayesian statistical method deduces how the prediction model will behave if a new data record or expert opinion is introduced. Bayesian statistics employs the Bayes theorem to reverse the direction of the dependencies, from the conditional probability of having a predicted output for a given data record, $\mathrm{P}(\mathrm{output}|\mathrm{data})$, to the probability of having a possible data record (e.g., new training set) for a predicted output, $\mathrm{P}(\mathrm{data}|\mathrm{output})$.

An example of Bayesian statistics is Helmenstine's (2017) drug test analysis.



In this example, shown in the related figure, $\{\mathrm{U}, \overline{\mathrm{U}}, +, -\}$ stand for a drug user, non-user, positive drug test result, and negative drug test result, respectively. Assume 0.5% of the training set are drug users (P(U) = 0.5%) and the probability that a drug test will be positive when taken by a drug user is 99% ($\mathrm{P}(+|\mathrm{U}) = 99\,\%$) (meaning the test will be negative for 1% of drug users ($\mathrm{P}(-|\mathrm{U}) = 1\,\%$). All other conditional probabilities are reported in the figure on drug testing analysis.

What will be the probability that a new data record (i.e., a new person added to the training set) with a positive drug test result is actually a drug user?

$$\mathrm{P}(\mathrm{U}|+) = \frac{\mathrm{P}(+|\mathrm{U})\mathrm{P}(\mathrm{U})}{\mathrm{P}(+)}$$
$$\mathrm{P}(\mathrm{U}|+) = \frac{\mathrm{P}(+|\mathrm{U})\mathrm{P}(\mathrm{U})}{\mathrm{P}(+|\mathrm{U})\mathrm{P}(\mathrm{U}) + \mathrm{P}(+|\overline{\mathrm{U}})\mathrm{P}(\overline{\mathrm{U}})}$$
$$\mathrm{P}(\mathrm{U}|+) = \frac{0.99 \cdot 0.005}{0.99 \cdot 0.005 + 0.01 \cdot 0.995} = 33.2\%$$

Since the probability is only 33.2%, it implies that even if a drug test produces a positive result, it is more likely the person is not a drug user. This result applies if we assume the person has the same prior probability as the general population (i.e., $\mathrm{P}(\mathrm{U}) = 0.5\%$). However, if we know more about a specific person (e.g., used drugs in the past, has a medical condition that makes recreational drug use dangerous), the prior changes, as does the posterior prediction. Hence, knowledge of the posterior is

critical and represents our best knowledge. This is important because in machine learning, the training data represent our best knowledge about a dataset, and it is crucial to be certain that a dataset is as accurate as possible. This is why assurance of data quality is very important.

The drug test example shows how prior probability $\mathrm{p}(\mathrm{U})$ is adjusted according to the posterior probability $\mathrm{P}(\mathrm{U}|+)$ defined by the model output (i.e., positive test result). This can be the result of designing a classifier to predict the occurrence of an output for a new training set. This is the main idea behind the Naïve Bayes classifier for categorical data of independent random variables.

Summary

Data science is a multidisciplinary field that has borrowed aspects from statistics, pattern recognition, computer science, and operational research. It derives information from data and applies it for predictive purposes. The use of the extracted information depends on the particular application, but in general, it aids an organization's decision-making processes.

There are wide-ranging applications of data science in different fields with respect to available data. These applications include industrial processes, organizational workflow, image data, text data, and broadcasting data. Typical sources of data are trademarked companies, governments, academic institutions, Web pages, and media platforms. These data are either quantitative or qualitative, and they are shaped (formatted) as structured, unstructured, or semi-structured. There are five characteristics to be taken into consideration when handling data: volume, variety, velocity, veracity/validity, and value of the input data.

The probability theory and its basic distributions are applied to large datasets to obtain a descriptive sample of the data, and then Bayesian statistics is employed to predict new data values.

Knowledge Check

Did you understand this unit?

Now you have the chance to test what you have learned on our Learning Platform.

Good luck!

Unit 2

Use Cases and Performance Evaluation

STUDY GOALS

On completion of this unit, you will have learned ...

... the importance of a use case for business.

... how to identify use cases.

... the steps to develop a predictive model for a specific use case.

... the metrics to evaluate the performance of a predictive model.

... the role of KPIs in business-centric evaluation.

... the different cognitive biases which influence the decision-making process.

2. Use Cases and Performance Evaluation

Introduction

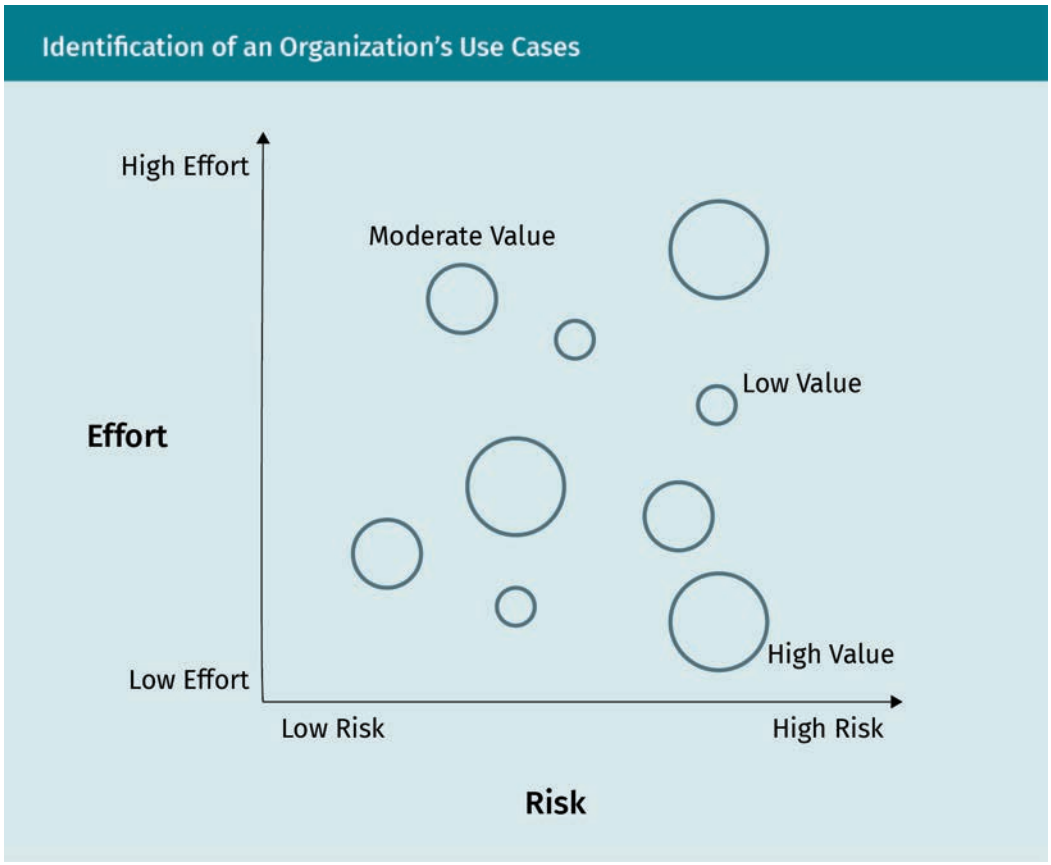
Pivotal Data Labs planned to use data science to understand TV show viewer habits and preferences to help with the formulation of the next series of these TV shows. According to a case study published by Pivotal, the challenge was handling the huge amount of broadcast data, much of which had to be manually collected. Therefore, Pivotal built an in-database data processing model that incorporates all data into a simple format that is easy to navigate. The value of the model developed by Pivotal the clear perspective on what variables affect a show's popularity, as well as viewer preferences for a certain show over the lifespan of that show. The model helps broadcasting clients make informed decisions based on a robust approach that was previously unavailable.

Use cases and scenarios show business managers the benefits data science tools can offer for enhancing decision-making within their organizations. Once a data science use case (DSUC) is identified and a prediction model developed, the performance of the model must be evaluated and its effectiveness in helping businesses reach their goals measured. But, attention must also be paid to the cognitive and motivational biases that might impact model inputs, thus influencing the model's performance and, consequently, an organization's decision-making. All these aspects will be discussed in this unit.

2.1 Data Science Use Cases (DSUCs)

Identifying DSUCs and Their Value Propositions

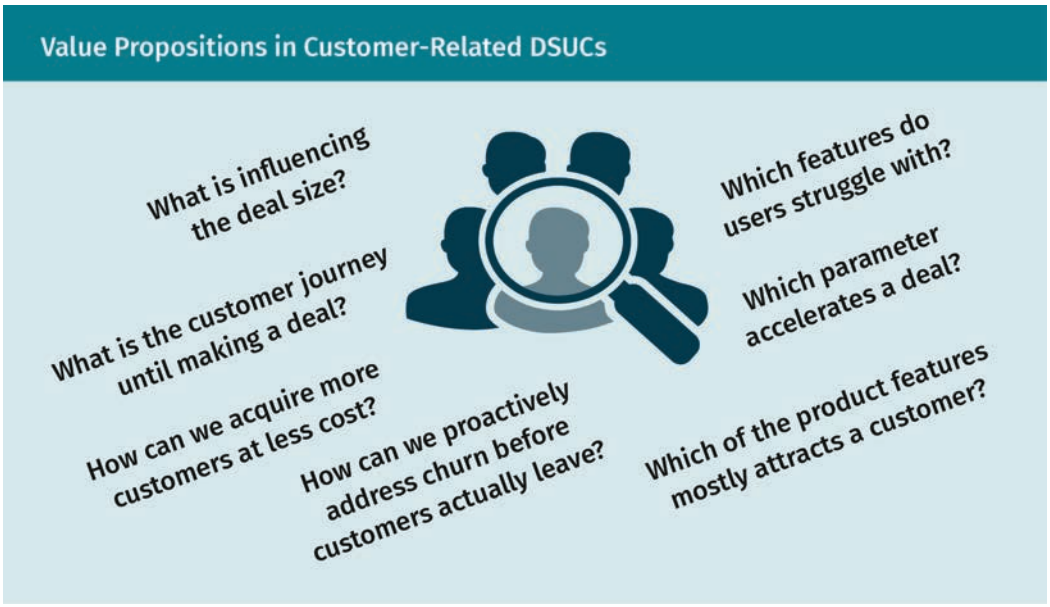
The real value of a business is unlocked through in-depth investigation of its collected data. Organizations must not only overcome the challenges inherent in managing this data but also identify the right DSUC for their business objectives. A well-matched DSUC will give managers valuable insights for addressing business challenges and improving future gains. The DSUCs which apply prediction techniques to extract value from collected data can vary widely across organizations. In general, however, a DSUC in any business is identified through three main aspects: achieved value, effort, and risk. Managers tend to measure a potential new project by how much it could improve their operational business and/or bottom line. The analysis of an organization's business should therefore focus on increasing gain, reducing risk, and/or decreasing effort.

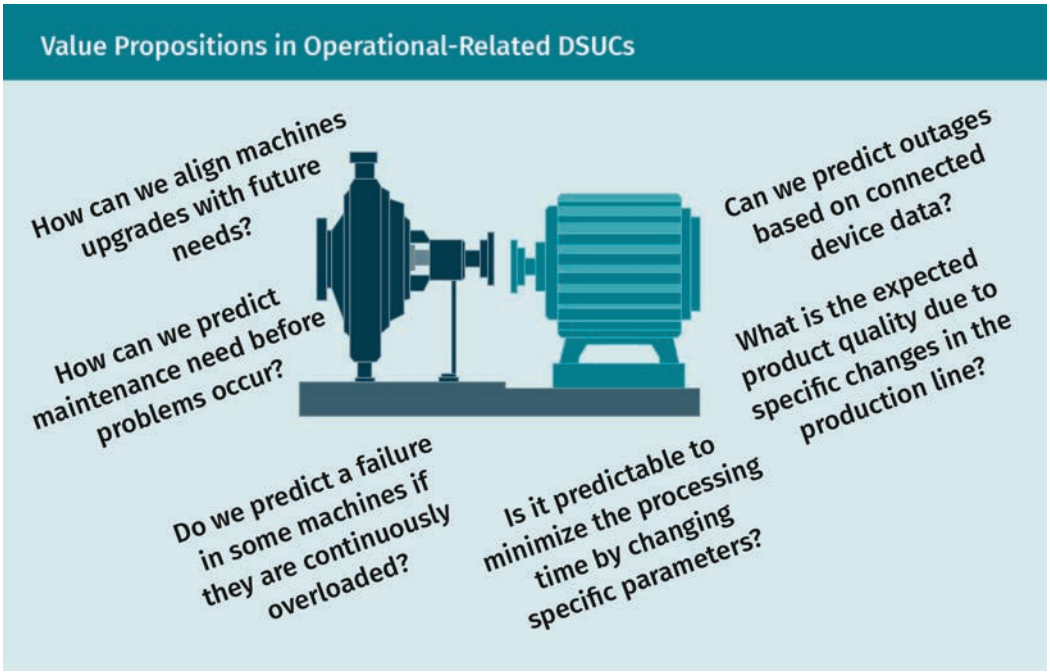


Every organization has to identify the kind of use cases to be tackled and ensure that the relevant datasets are available. In addition, they need to answer the following questions:

- What is the value of the knowledge gained from applying data science tools to the dataset?
- What will be learned about the dataset?
- What will be learned about the hypothesis the data science tools will test?
- What will be the value of that knowledge if the prediction model developed shows good business performance? If it shows a negative business outcome?

For most organizations, the value propositions of applying data science tools can be summarized using three figures (Datameer, 2016).


Value Propositions in Customer-Related DSUCs
What is influencing the deal size?
What is the customer journey until making a deal?
How can we acquire more customers at less cost?
How can we proactively address churn before customers actually leave?
Which features do users struggle with?
Which parameter accelerates a deal?
Which of the product features mostly attracts a customer?


Value Propositions in Operational-Related DSUCs
How can we align machines upgrades with future needs?
How can we predict maintenance need before problems occur?
Do we predict a failure in some machines if they are continuously overloaded?
Is it predictable to minimize the processing time by changing specific parameters?
Can we predict outages based on connected device data?
What is the expected product quality due to specific changes in the production line?

Learning the Dataset and Building the Prediction Model

Once a DSUC is identified, the relevant dataset needs to be collected from the available data sources (or new data sources built). Types of data sources include internal/external databases, sensor data, static files, and Web scraping. The data collection process may be expensive if humans are required to study the data and manually insert important tags, labels, and/or valuable comments.

The collected dataset is forwarded to preprocessing where it is scrubbed of any noise and scanned for redundant records and missing values. Cleaning data typically requires significant domain knowledge to make decisions about the best method to deal with errors in the data. At the end of the preprocessing stage, the representative size of the dataset will have been reduced and misleading information removed. A dataset will contain variables (i.e., features) with numerical, categorical, and/or textual values. Certain features may not be relevant to or compatible with the desired DSUC values. Therefore, features should be carefully selected such that they can be used to determine output value propositions.

Now that the data have been readied, it is time to build the prediction model. The purpose of the model is to define the relationships between the inputs (selected relevant features) and outputs (DSUC value) of the dataset. This is accomplished by establishing mathematical functions between the inputs and outputs. The dataset is divided into two sets of data records: training and testing. The training set is used to build and learn the numerical model, and the testing set is used to evaluate the model's accuracy. The model that is developed will provide the current DSUC value of the dataset and also predict the DSUC value if a new scenario with different data records is examined. There may be a chance the model will need updating, especially if there are changes in the datasets or new data records added.

There are many numerical approaches which have been established for learning datasets and building robust prediction models, namely machine learning approaches. These approaches depend on the nature of the dataset's outputs. Classification

approaches are used if the outputs are to be categorized into classes (e.g., {sunny, windy} for weather datasets), and regression approaches are used if the outputs are probability density distributions (e.g., {profits} for customer purchasing datasets).

Making Predictions and Decisions

Once the prediction model has been built, it is ready to find the function which relates selected features of the data (inputs) to the objective value of the DSUC (output). Iterations of the model may need to be run several times until it produces a reasonably high level of accuracy with respect to the testing set. The output of a prediction model is either a probability (for a classification model) or a probability density distribution and/or a number with a degree of uncertainty (for a regression model). For classifications, determining the model's accuracy requires a threshold to be set on the model's output (e.g., transactions with a probability higher than 80 percent are flagged as fraud). For regressions, an optimal point estimator needs to be determined from a predicted probability distribution (e.g., error between predicted output and target output is less than five percent).

The DSUC value is presented to the end user (e.g., a manager) so they can determine the corresponding action to be taken or decision to be made. In some cases, the model itself (rather than its output) is submitted to the end user. In this case the end user must decide how to translate the probabilities produced by the model into actions and thresholds. Therefore, to make it easier for the end user, the model must be user-friendly with an intuitive, front-end interface. The end users will decide how to align the model's outputs with the overall business objectives and project goals. For example, should the fraud prediction be accepted if it is higher than 80 percent but less than 90 percent? Is it acceptable to have a model with more false positive outputs but fewer false negative outputs? These are the trade-offs that need to be made and the parameters that need to be determined so end users can effectively benefit from the model's outputs. Different choices in trade-offs and parameters will permit different business scenarios to be considered.

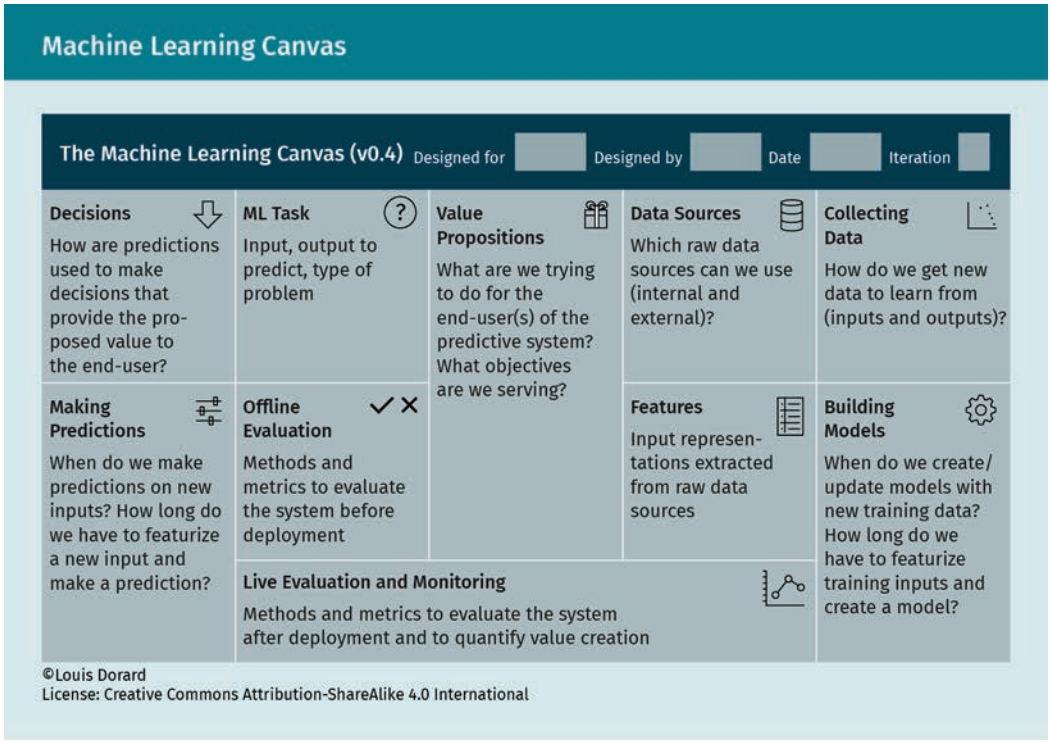
For example, in the case of fraud analysis, the outputs of the prediction model are fraud probabilities. The decision about which model threshold to choose is a strategic one that will impact the identification of fraud/non-fraud cases. A low threshold will result in more cases being incorrectly identified as fraud, while a high threshold will result in the under-identification of actual fraud cases. Neither scenario is perfect—both will have incorrectly-identified cases. The question becomes which threshold is preferable for a specific use case?

Sometimes the end user will make a decision that impacts the data records. For example, suppose product price is one of the selected features in a dataset with which a model was built, and at some point management decides to change the price of a product. To be prepared for such an event, the model should be developed such that it includes a feedback loop that can accommodate these types of changes, and the model can be retrained accordingly.

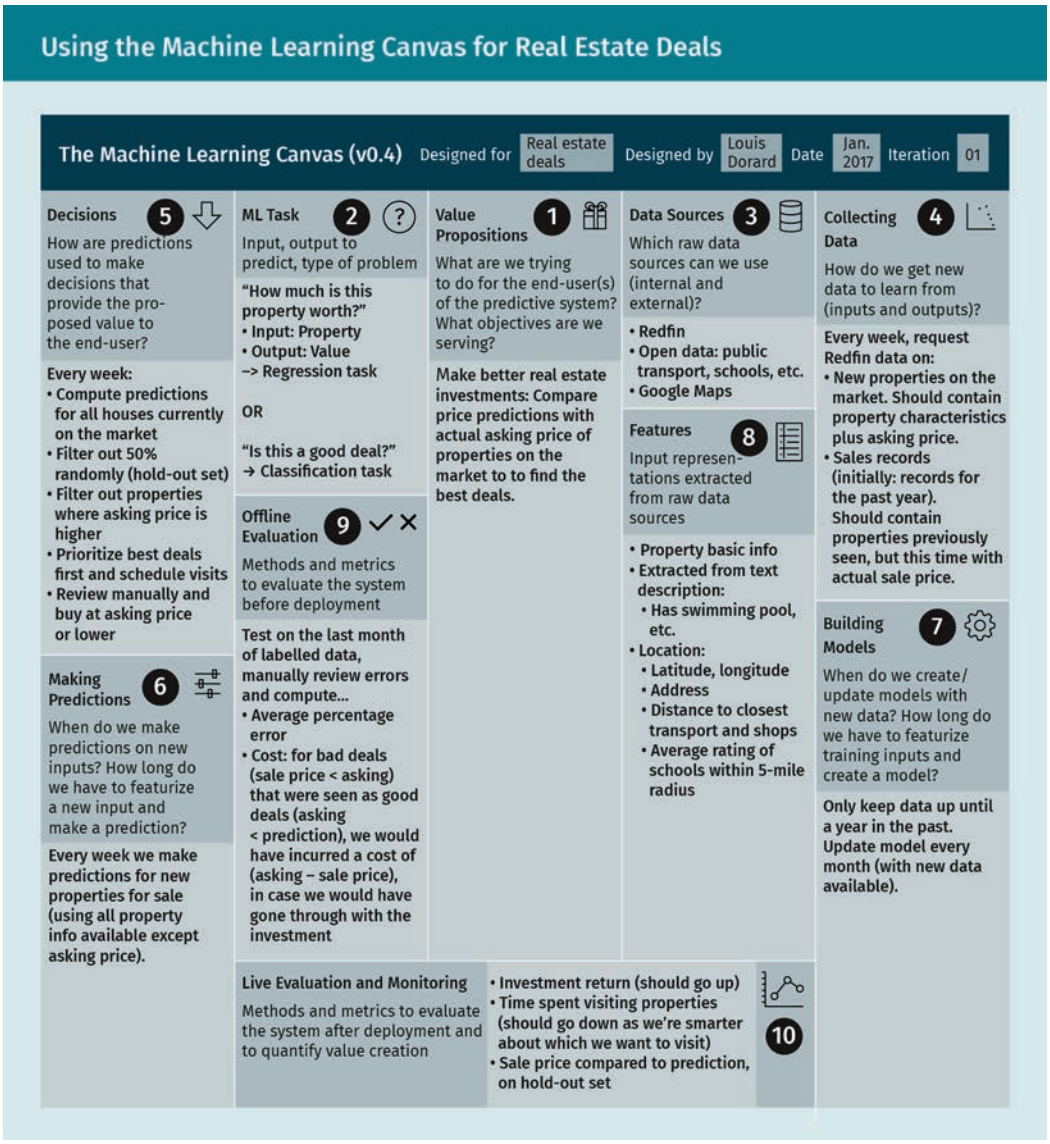
The ultimate goal when building an intelligent model is to completely automate end users' corresponding decisions. Decisions are often based on the model's confidence in its predictions. For example, suppose a model is built to predict whether a hotel review is real or fake. If the model is confident about its predictions, the review can be automatically accepted or rejected without necessary contact with an end user.

Machine Learning Canvas

A helpful tool in identifying use cases is the "machine learning canvas," developed by Dorard (2017) to provide business managers with a user-friendly work procedure.



The tool collects in one place all the steps required to identify a use case and achieve its value proposition. For example, we can use the machine learning canvas in the case of real estate deals. Our value proposition is to make more lucrative real estate investments by comparing the price predictions for properties with their actual asking prices in order to identify the best deals. The procedure of the analysis using Dorard's machine learning canvas is summarized in the following figure.



2.2 Performance Evaluation

The overall evaluation of how well a DSUC has been modeled and its predictive values applied successfully within a business can be divided into two parts. The first part involves evaluating the developed prediction model and measuring its performance through a list of known numerical metrics. The second part involves evaluating how the model's outputs (i.e., the DSUC values) are used to better understand and improve a business. The latter is usually accomplished using a defined list of key performance indicators (KPIs).

Model-Centric Evaluation: Performance Metrics

In this section we discuss several metrics for measuring how well a prediction model performs its classification or regression task.

Classification model evaluation metrics
For a classification problem with two output classes {“yes”, “no”}, the output of the prediction model is a probability that—depending on the set threshold—determines which class the output is assigned to. There are four possible results of a classification prediction model when applied to a data record: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). For a TP, the classifier labels a “yes” data record as “yes,” resulting in a correct prediction. For a TN, the classifier labels a “no” data record as “no,” resulting in a correct prediction. In the case of an FP, the classifier mistakenly labels a “no” data record as “yes,” resulting in a type l classification error. And in the case of an FN, the classifier mistakenly labels a “yes” data record as “no,” resulting in a type ll classification error. This list of possible results defines an important metric for the model, called the confusion matrix.

Confusion Matrix			
		Model Output	
		YES	NO
Desired Output	YES	Count of TPs	Count of FNs
	NO	Count of FPs	Count of TNs

When evaluating the model based on these four possible outputs, the metrics used are accuracy, precision, and recall.

Accuracy is the ratio of the number of correct predictions to total predictions.

$$Accuracy = \frac{\text{count of } TP + \text{count of } TN}{\text{count of } TP + \text{count of } TN + \text{count of } FP + \text{count of } FN}$$

Precision measures how correct the model is when returning a positive result.

$$Precision = \frac{\text{count of } TP}{\text{count of } TP + \text{count of } FP}$$

Recall measures how often the model produces true positives. The recall metric is used if we are more tolerant of false positives (e.g., a healthy person is misdiagnosed with an illness) than false negatives (e.g., a sick patient dies because of a misdiagnosis).

$$Recall = \frac{\text{count of } TP}{\text{count of } TP + \text{count of } FN}$$

The classification model may apply a threshold (i.e., cutoff) to the output to distinguish between the two classes {"yes", "no"}. For example, if the cutoff is set to 80 percent, the model will assign a data record to the "yes" class if the model produces an output higher than 80 percent. Otherwise, it will assign this data record to the "no" class. Hence, by altering the cutoff value, we can obtain different results from the model. As a consequence, the numbers of TP, TN, FP, and FN change.

The receiver operator characteristic (ROC) curve displays the trade-off between the true positive rate and the false positive rate at every possible cutoff value. An ideal model is the one which is able to classify the testing set with a 100 percent TP rate and 0 percent FP rate. Therefore, the ROC curve helps find the best possible realistic cutoff value which results in the highest TP rate and lowest FP rate. An ROC curve can be generated using the following steps:

1. Choose a cutoff value between [0: 100] percent of the maximum value of the model output.
2. Assign the testing set according to their classes and count the TP, TN, FP, and FN values.
3. Calculate:

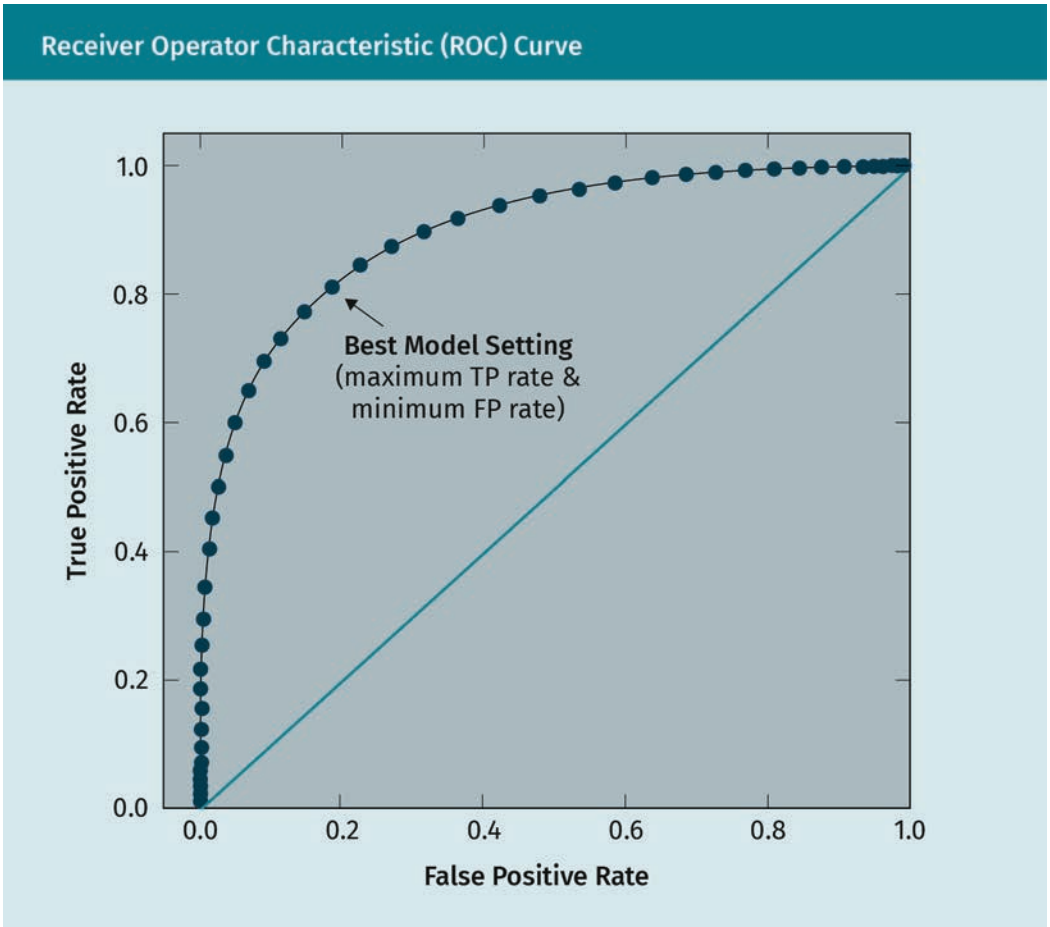
$$\text{False positive rate} = \frac{\text{count of } FP}{\text{count of } FP + \text{count of } TN}$$

and

$$\text{True positive rate} = \frac{\text{count of } TP}{\text{count of } TP + \text{count of } FN}$$

4. Form a single point on the ROC curve with the coordinate: (false positive rate, true positive rate).
5. Choose another cutoff value and repeat steps 2 to 4.

In the ROC curve given, we notice that the closer the curve to the upper left corner, the more efficient the model (i.e., very close to the ideal model). This can be measured by calculating the area under the curve (AUC), where the AUC for an ideal model is 1.



Regression model evaluation metrics

The output of a regression prediction model is a probability density distribution which must be translated into a number (i.e., optimal point estimator) in order to be useful operationally. Therefore, for the testing set, the evaluation of such a model simply focuses on how close the model output (y) is to the desired output (d). The typical metrics used to evaluate the performance of a regression model are absolute error; relative error; mean absolute percentage error; square error; mean square error; mean absolute error; and root mean square error.

Absolute error is the absolute difference between the model's output and the desired output.

$$\varepsilon = |d - y|$$

Relative error normalizes the calculated absolute error with respect to the desired output to obtain a unit-less percentage. This is important because the absolute error is meaningless without a sense of the units involved.

$$\varepsilon^* = \left|\frac{d - y}{d}\right| \cdot 100\,\%$$

The relative error may not be quite representative, especially for small numbers. In this case, d has to be greater than zero in all cases, or the operation is not valid.

Mean absolute percentage error (MAPE) is the average relative error calculated over the entire testing set of n data records.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{d_i - y_i}{d_i}\right| \cdot 100\,\%$$

MAPE is very useful if the underlying probability density distribution of the values is "sufficiently far" from zero, such that zero does not have a significant impact.

Square error (i.e., the squaring of the error) ensures that a positive quantity is obtained and adds significant weight to the large error values if they take place at some records of the testing set. For example, for two testing records ($\varepsilon_1 = 1$ and $\varepsilon_2 = 2$), squarring them (${\varepsilon_1}^2 = 1$ and ${\varepsilon_2}^2 = 4$) gives more weight to the higher error ε_2.

$$\varepsilon^2 = (d - y)^2$$

Mean square error (MSE) is the average square error over the entire testing set for n data records.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(d_i - y_i)^2$$

The MSE can be dominated by outliers depending on the underlying probability density distribution.

Mean absolute error is more robust than the mean squared error with respect to data-sets containing outliers.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|d_i - y_i|$$

Root mean square error is the square root of the mean square error and produces a result with a magnitude that is easier to interpret and on the same scale of the desired and predicted outputs.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - y_i)^2}$$

Business-Centric Evaluation: The Role of KPIs

After the prediction model has been successfully evaluated using the above-mentioned evaluation metrics, it is ready to be implemented to produce the DSUC value for the associated business problem. Meanwhile, the end user (i.e., decision-maker) should be able to measure whether the DSUC value has been successfully implemented in their business. This evaluation is conducted using a list of key performance indicators (KPIs) to reflect how many business objectives have been achieved. Most KPIs will focus on measurements related to improving revenue, reducing costs, increasing efficiency, and/or enhancing customer satisfaction.

Characteristics of effective KPIs
For a KPI to be truly helpful as a metric for determining business improvements and achievement of business objectives, it should be

1. easy to comprehend and simple to measure;
2. comprised of small, measurable elements (e.g., number of daily operations, amount of daily production, employee workload);
3. assigned to the appropriate, relevant task manager;
4. able to indicate positive/negative variations from the business objective;
5. achievable within the resource constraints (e.g., staff, machines, processes);
6. defined with both start and end dates for measuring; and
7. visible across the entire organization.

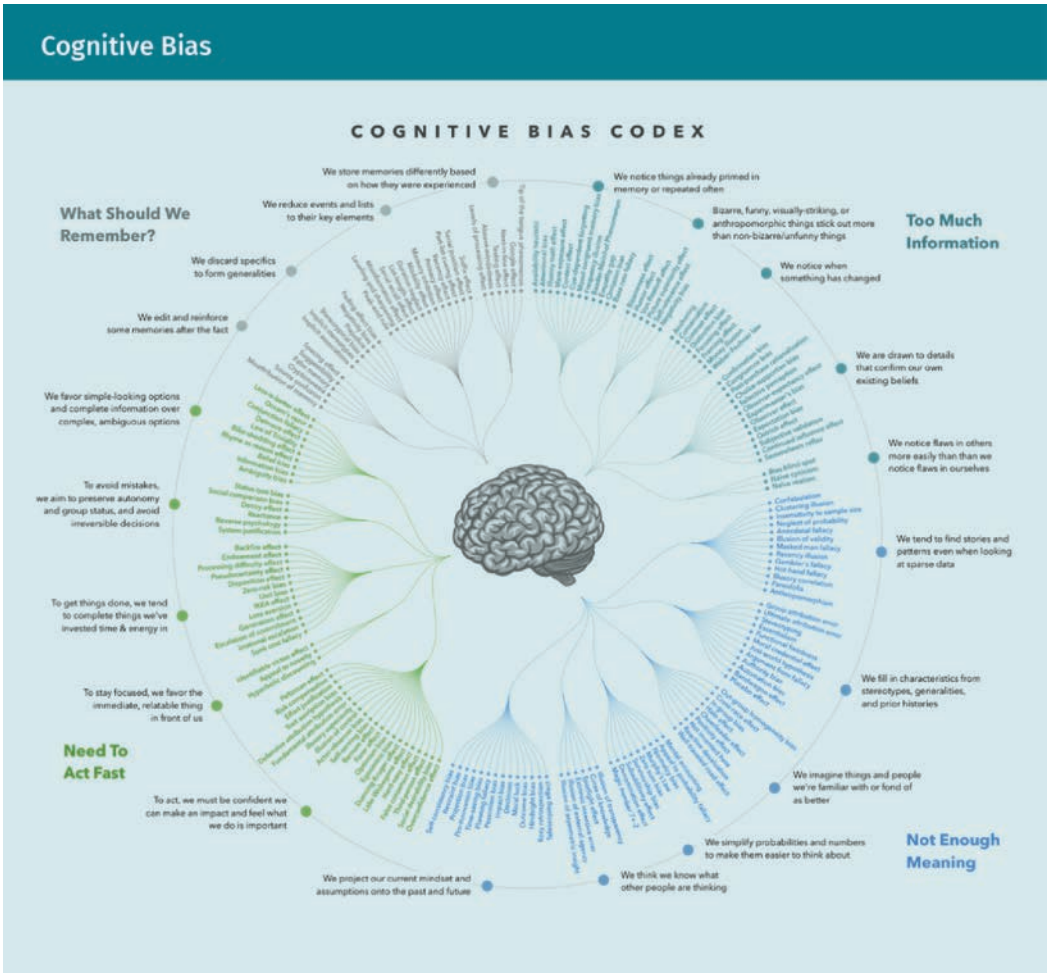
Examples of KPIs
Some examples of effective KPIs which are routinely implemented in organizations to measure DSUC performance include the following:

- growth/shrinkage per year
- time to finish a task
- percentage of tasks completed within a specified time
- cost of service delivery
- machine downtime and availability
- number of complaints
- workload of staff
- revenue per employee
- production yield
- employee/customer satisfaction index

Once a KPI is defined, we have to determine the best method to assess performance against it. It may be helpful to divide the assessment into more manageable elements and measure each separately.

Cognitive Biases and Decision-Making Fallacies

Montibeller and Winterfeldt (2015, p.1230) reported that "[b]ehavioral decision research has demonstrated that judgments and decisions of ordinary people and experts are subject to numerous biases." In data science, the cognitive and motivational biases may be disruptive not only to the collected dataset but also at different stages of data processing. These biases seriously influence the quality of the developed prediction model. As a result, decision-making may be inaccurate. There are countless biases that are difficult to avoid, but being aware of them allows the data scientist to take them into consideration.



Relevant cognitive biases

The list provided below is for the common cognitive and motivational biases that can greatly distort prediction model inputs.

Common Cognitive and Motivational Biases	
Bias	Description
Anchoring	• "occurs when the estimation of a numerical value is based on an initial value (anchor), which is then insufficiently adjusted to provide the final answer."
Affect influenced	• "occurs when there is an emotional predisposition for, or against, a specific outcome or option that taints judgments."
Ambiguity aversion	• "People tend to prefer gambles with explicitly stated probabilities over gambles with diffuse or unspecified probabilities."
Equalizing bias	• "occurs when decision makers allocate similar weights to all objectives."
Confirmation	• "occurs when there is a desire to confirm one's belief, leading to unconscious selectivity in the acquisition and use of evidence."
Base rate fallacy	• "People tend to ignore base rates when making probability judgments and rely instead on specific individuating information."
Desirability of options	• "This bias leads to over- or underestimating probabilities, consequences in a direction that favors a desired alternative."
Insensitivity to sample size	• "People tend to ignore sample size and consider extremes equally likely in small and large samples."

De-biasing techniques

De-biasing techniques attempt to eliminate, or at least reduce, the effect of the cognitive and motivational biases and avoid any related strategy- and association-based errors.

De-biasing Techniques	
Bias	De-biasing Techniques
Anchoring	• "Avoid anchors • Provide multiple and counter anchors • Use different experts who use different anchors"
Affect influenced	• "Avoid loaded descriptions of consequences in the attributes • Cross-check judgments with alternative elicitation protocols when eliciting value functions, weights, and probabilities • Use multiple experts with alternative points of view"
Ambiguity aversion	• "Model and quantify ambiguity as probability distribution • Model as parametric uncertainty (e.g., over the bias parameter of a Bernoulli process) or secondary probability distribution"
Equalizing bias	• "Rank events or objectives first, then assign ratio weights • Elicit weights or probabilities hierarchically"
Confirmation	• "Use multiple experts with different points of view about hypotheses • Challenge probability assessments with counterfactuals • Probe for evidence for alternative hypotheses"

Bias	De-biasing Techniques
Base rate fallacy	• "Split the task into an assessment of the base rates for the events and the likelihood or likelihood ratio of the data, given the events"
Desirability of Options	• "Use analysis with multiple stakeholders providing different value perspectives • Use multiple experts with different opinions • Use incentives and adequate levels of accountability"
Insensitivity to sample size	• "Use statistics to determine the probability of extreme outcomes in samples of varying sizes • Use the sample data and show how and why extreme statistics are logically less likely for larger samples"

Summary

Use cases are important in many organizations and fields for obtaining vital value propositions. The objective of applying data science in such areas is to improve output value while lowering risk and decreasing the effort required to complete a task. After identifying a data science use case, the corresponding dataset which contains the relevant characteristics is collected. This dataset is put through preprocessing, during which any noise and missing values are handled. Afterwards, a list of the most relevant data features are selected and employed to build the prediction model. There are two task options for a prediction model: classification or regression.

Metrics are determined to evaluate the performance of the developed classification and regression prediction models. The model which meets performance requirements is applied to the dataset, and the output value is forwarded to decision-makers who use it to inform their actions. Then, using a specified group of key performance indicators, these decision-makers measure how successful the application of the output value to their business operations has been.

Since the whole task is performed through human interaction, some cognitive and motivational biases may influence the inputs of the prediction model and therefore affect the output. These biases should be carefully avoided.

Knowledge Check

Did you understand this unit?

Now you have the chance to test what you have learned on our Learning Platform.

Good luck!

Unit 3

Data Preprocessing

STUDY GOALS

On completion of this unit, you will have learned ...

... data transmission methods and techniques.

... how to handle missing values and outliers in a dataset.

... how to apply correlation analysis.

... data transformation approaches.

... data visualization tools.

3. Data Preprocessing

Introduction

Swedish energy company Vattenfall collected weather observations every minute for 274 days. The data scientists noticed several errors in the raw dataset:

- missing values
- an outlier in a wind speed observation of 171 m/s at time = 15:23, which was 3.163 m/s one minute before and 3.148 m/s one minute after
- an outlier in an ambient temperature observation of 61.757 C° at time = 17:54, which was 24.793 C° one minute before and 24.677 C° one minute after

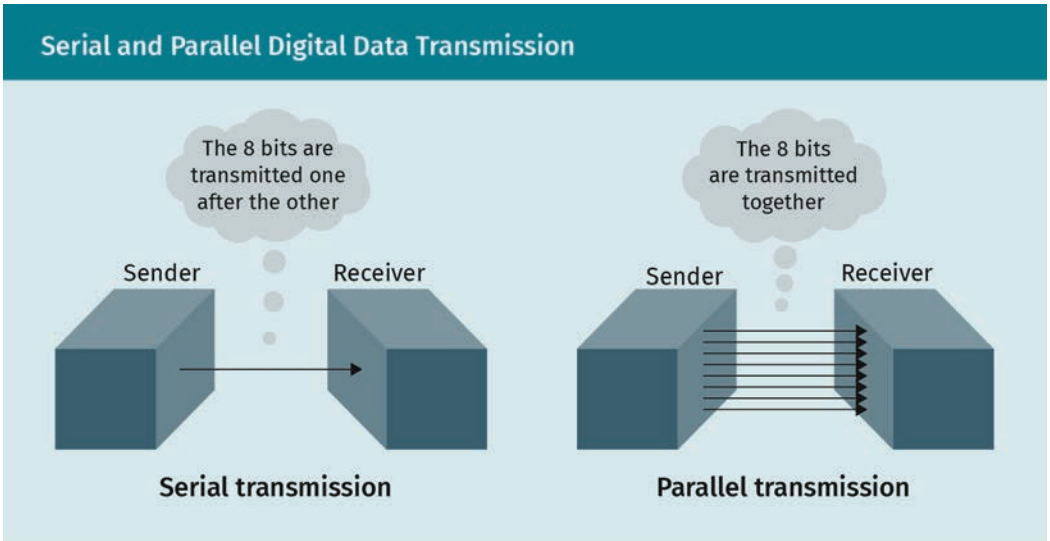
There are two important activities taking place in this case study. The first activity is transmitting the weather data from the sensors and other devices to the data scientists' computers. The second activity is cleaning errors in the dataset. These errors are usually in the form of missing data values and/or odd values that may not be true. The judgment typically requires detailed domain knowledge and a specialist to decide the reasons for the missing and/or odd values. For example, is there a problem with the sensor? Maybe the sensor is about to malfunction?

Furthermore, two "hidden" activities are necessary to complete the process. The first hidden activity is transforming variables with different scales into variables with one unique scale so that all data variables carry the same weight. The second hidden activity is visualizing the data variables to help with discovery of data errors and variable correlations. These four activities will be the main scope of this unit.

3.1 Transmission of Data

Data sourced from different departments in an organization may need to be transmitted in order to create the business-related dataset. For example, an industrial organization has machines that operate according to a defined shift schedule and predetermined employee workloads and production capacities. Sensors are attached to these machines so that important parameters, e.g., idle and processing times, can be measured. To apply data science in this organization, all data variables from the machines must be brought together to form a dataset. These variables are transmitted from the attached machines' sensors (for the measured parameters), the HR department's documents (for the workload), and the production manager (for the operation details).

Transmission of the required data can be accomplished manually or electronically. Manual transmission is the simplest form and is performed by manually inserting variables into the dataset. For digital data in the form of bits (computer memory units), electronic transmission is performed using local and/or wireless area networks. Electronic transmission uses serial or parallel transmission links. With serial transmission, digital data are sent bit by bit over one channel. With parallel transmission, multiple channels are used to deliver multiple data bits each time (Thakur, 2017).



Although parallel data transmission is much faster, it comes at a high cost and is limited to short transmission distances. Serial data transmission costs less, but the order of the data bits is important for dictating how they will be organized on the receiver side. However, this method is considered more reliable because a bit is only transmitted if its preceding bit has been delivered. Serial transmission is achieved using an asynchronous technique or synchronous technique.

Transmission: Asynchronous and Synchronous	
Asynchronous transmission	Synchronous transmission
• The bit stream has start and stop bits, with a variable period between transmissions. • The start bit tells the receiver to expect the transmitted stream, while the stop bit terminates this transmission. • It is cheap, but slow, with additional overhead for the start and stop bits.	• The bit stream is combined into longer frames with a constant period between transmissions. • Any gaps among the streams are filled with idle streams of bits of 0 or 1. • It is fast, with no additional overhead.

The data transmission rate is expressed in terms of the number of bits transmitted per second (bps).

3.2 Data Quality, Cleansing, and Transformation

A collected dataset with all variables and data records will usually have quality issues. These issues are due to values that are noisy, inaccurate, incomplete, inconsistent, missing, duplicate, or outlying. It is important to note that there are "true" outliers (data events that are real but appear far away from the bulk of the other data events) and "fake" **outliers** (the result of poor data quality). There are many approaches for dealing with data quality issues, and more than 80 percent of a data scientist's time is spent on this aspect. The prior in the Bayesian sense is the basis on which predictive models are built. If missing values and outliers are resolved, the prior changes, which alters the basis on which the machine learning model operates.

Outlier
An outlier is a data record which is seen as an exceptional and incomparable case of the input data.

Missing Values and Outliers

In some data records, there may be values which were not observed (i.e., missing values) or incorrectly observed (i.e., outliers) during collection. Several methods are routinely employed to resolve missing values and outliers.

1. Removal of the data records with missing values and/or outliers: This method is recommended for large datasets where the removal of some records will not affect comprehension of the data. This method can only be followed after it is confirmed that removing the chosen records will not influence the results. For example, under a rare condition, a sensor may be unable to deliver a value. In such a case, removing the record might lead to exclusion of the most interesting aspect of the dataset.
2. Replacement of the missing value or outlier with an interpolated value from neighboring records: For example, we have a dataset for daytime temperatures (time: {11:00, 11:01, 11:02}, temperature: {20 C°, x, 22.5 C°}) where x is a missing temperature value or an outlier (i.e., out-of-range value). The value of x is replaced by the linearly-interpolated value obtained from the temperatures preceding and proceeding it $\left(\mathrm{x} = \frac{22.5+20}{2} = 21.25\ \mathrm{C}^{\circ}\right)$.
3. Replacement of the missing value or outlier with the average value of its variable along all data records
4. Replacement of the missing value or outlier with the most-often observed value for its variable along all data records

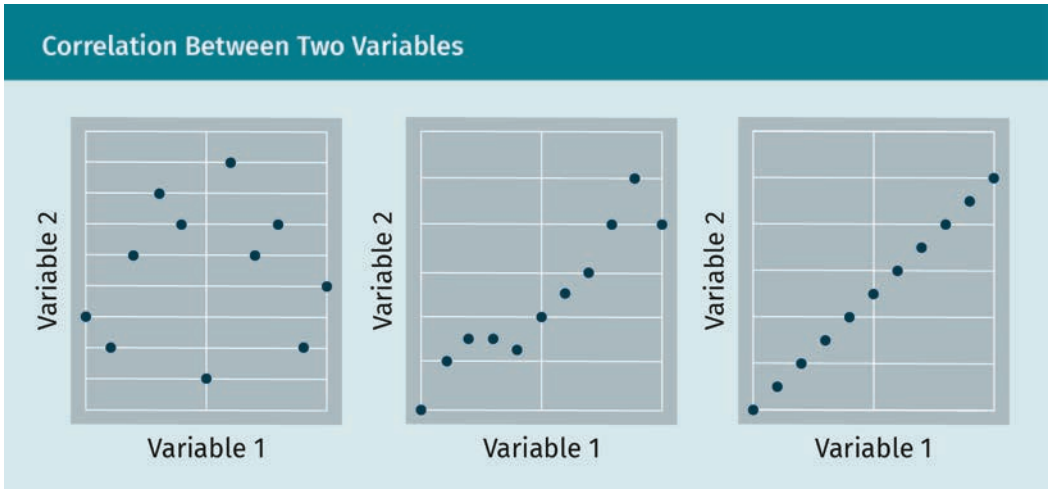
A new variable may be introduced to the dataset with a value of "0" for the normal data records and "1" for the data records containing missing and/or outlier values that were handled by one of the above methods. By doing so, we ensure the original information is not lost.

Duplicate Records

If there are duplicate records within the dataset, they are removed before proceeding with the data analysis in order to reduce computing time. However, if they are kept within the dataset, they will not degrade the outcomes of the analysis.

Redundancy

Other issues that may appear within the dataset are related to the existence of redundant and irrelevant variables. We resolve these issues by applying correlation analysis between each pair of variables, allowing us to remove those variables which show high correlation with respect to other variables without losing any important information about the dataset. The correlation between two variables can be seen in the following figure where the circle shape indicates that the variables are not correlated; the cigar shape indicates partial correlation; and the line shape indicates strong correlation.



The correlation coefficient (ρ) between two data variables x and y, is calculated as:

$$\rho(x, y) = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}}$$

where $\overline{x}$ and $\overline{y}$ are the average values of variables x and y, respectively, for a dataset of n records.

The correlation coefficient is a static measurement for the degree relationship between the two variables and ranges from –1 to 1. If ($\rho = 1$) the two variables are fully correlated, and if ($\rho = 0$) there is no indication of correlation or independent variables. Negative correlation coefficients imply that the variables are anti-correlated (i.e., negatively correlated), meaning that when x goes up, y goes down, and vice versa. We can set a threshold on the value of ρ, and if the correlation exceeds this threshold, one of the two variables can be removed from the dataset with negligible influence on performance.

Furthermore, we may apply one of the dimensionality reduction approaches, such as principal component analysis (PCA). PCA sorts the variables according to their importance, thereby removing the variables that have a minor influence on the data's variability. Such a technique results in a dataset with a fewer number of variables.

Transformation of Data

Data transformation is required to convert the dataset into a form suitable for applying data science. The main transformation methods are variable scaling, decomposition, and aggregation.

Data Transformation Methods

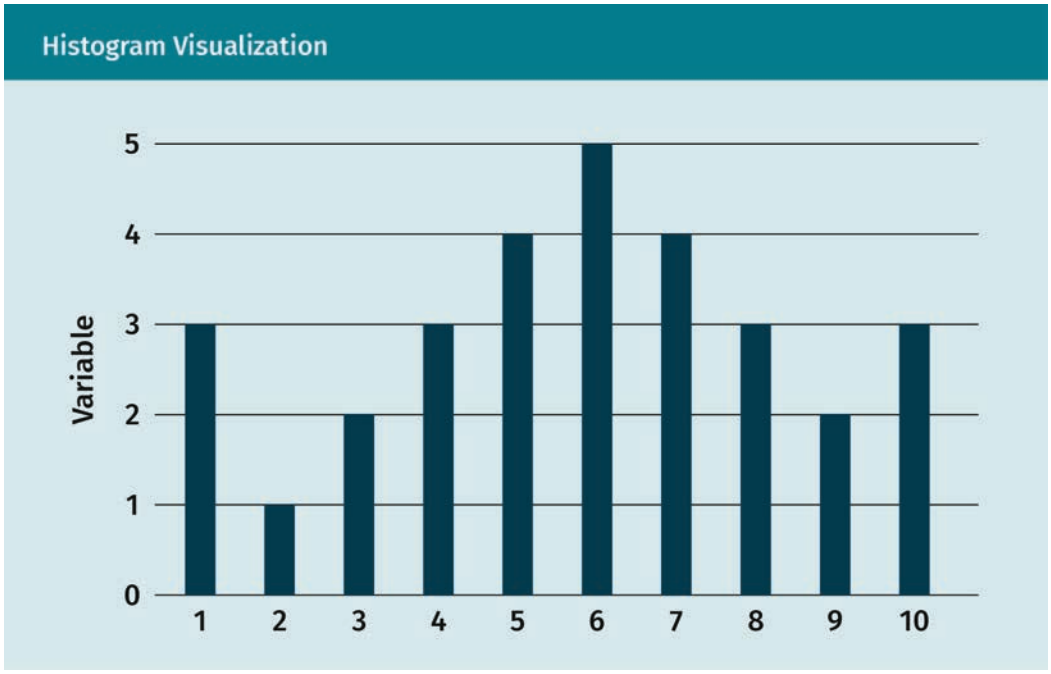
Transformation method	Description
Variable scaling	The dataset may include variables of mixed scales. For example, a dataset contains income values in dollars, number of purchases per month, and amount of car fuel consumed per month. The modeling techniques work on scaled variable values, e.g., between -1 and 1, to ensure that all analyzed variables are weighted equally. The scaling may be performed by normalizing a variable's value with respect to its maximum value. $$x_i = \frac{x_i}{\max(x)}$$ The other option is to remove the variable's average and divide by the variable's standard deviation. $$x_i = \frac{x_i - \overline{x}}{\mathrm{std}(x)}$$
Variable decomposition	Some variables may need to be further decomposed into more variables for better data representation. For example, a time variable may be decomposed into hour and minute variables. Furthermore, it may turn out that only one of the two variables (hour or minute) is relevant, so the irrelevant variable is removed from the dataset.
Variable aggregation	Alternatively, two variables may be more meaningful if they are merged (i.e., aggregated) into one variable. For example, "gross income" and "paid tax" variables may be aggregated into one variable, "net income."

3.3 Data Visualization

Data visualization is adopted to enhance understanding of a dataset via graphical representation (Runkler, 2012). Visualized data are easier to analyze. Common data visualization types include histograms; scatter plots; geomaps; charts (area, bar, pie, combo, and bubble); and heat maps.

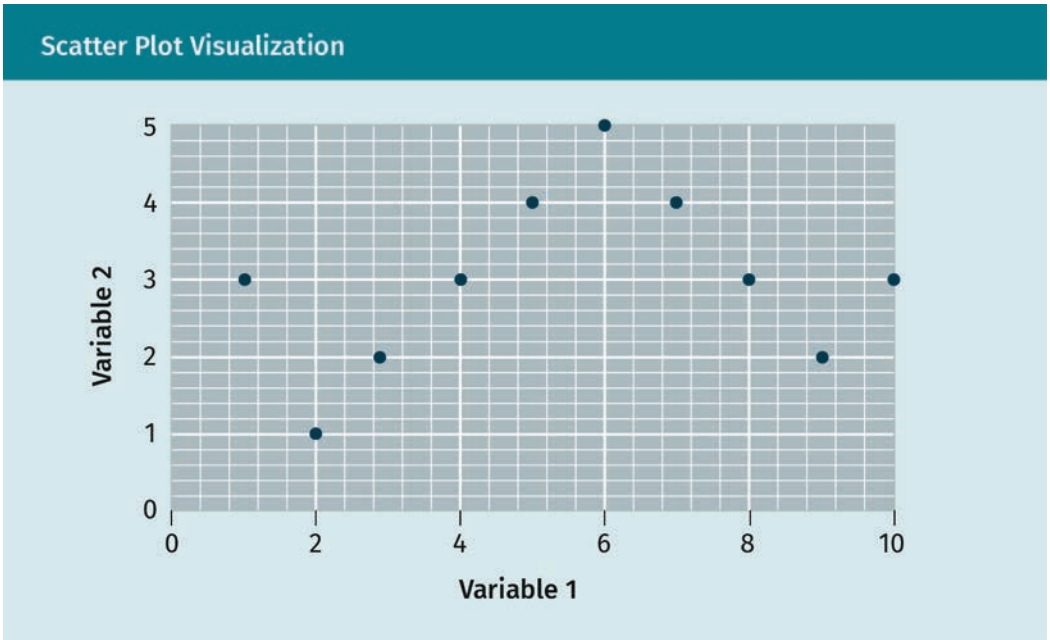
Histogram

A histogram is a graphical display of one variable using different bar heights.



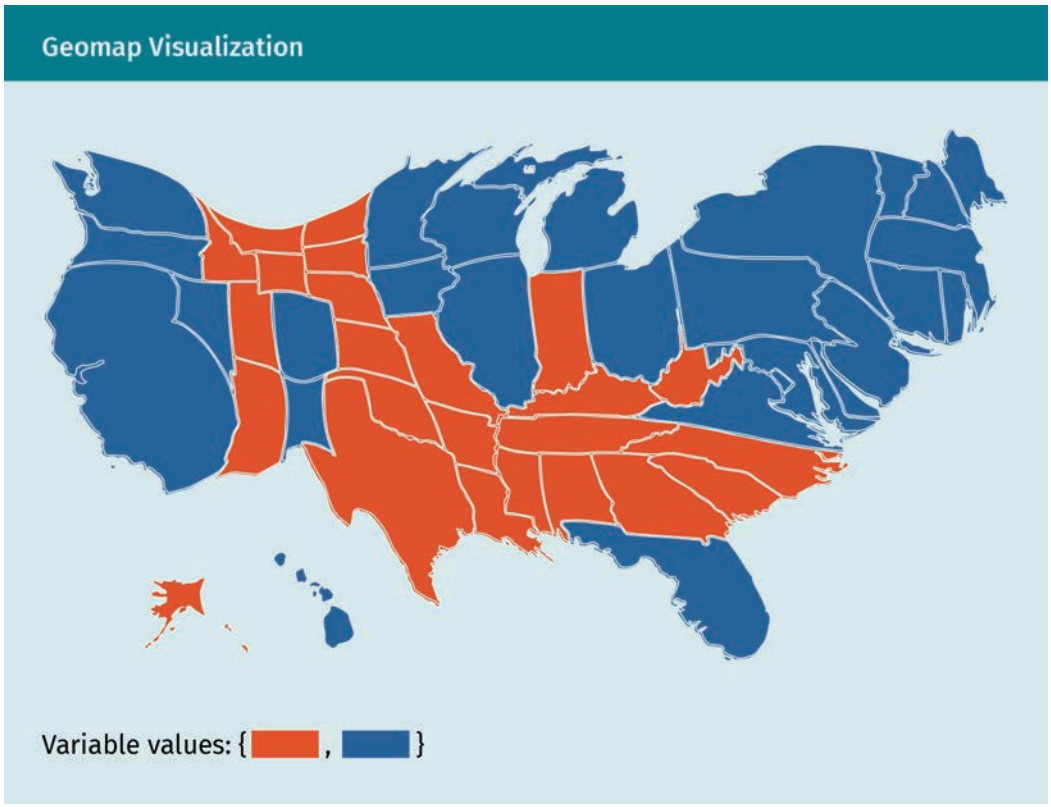
Scatter Plots

The scatter plot is a graph with two axes on which two variables are plotted to show their correlation.



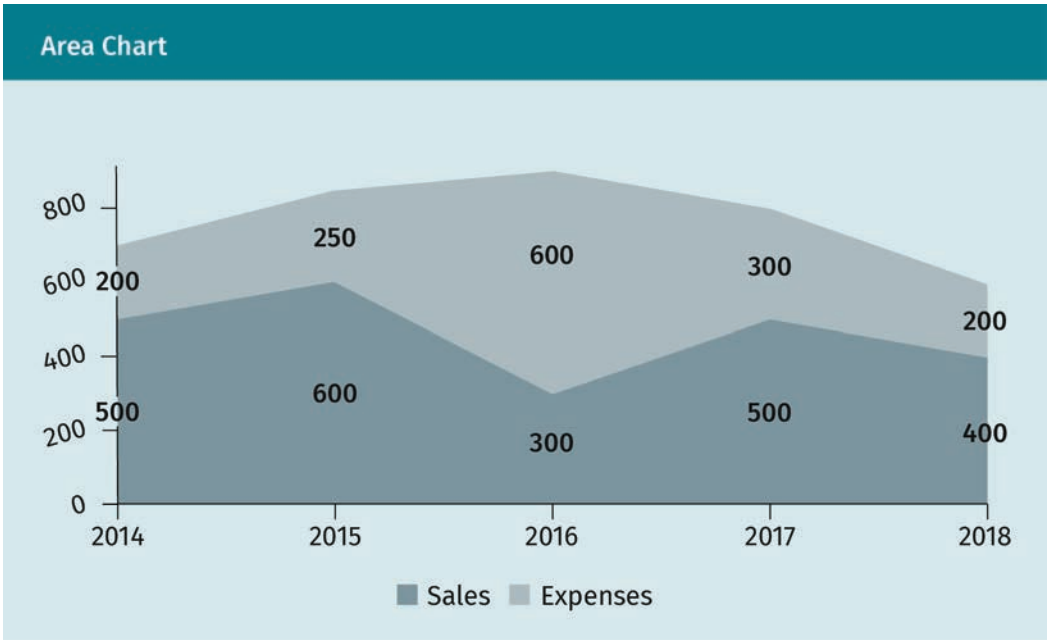
Geomaps

A geomap is a geographical map of continent(s), region(s), and/or countries on which the variables' values are displayed using a color scale.



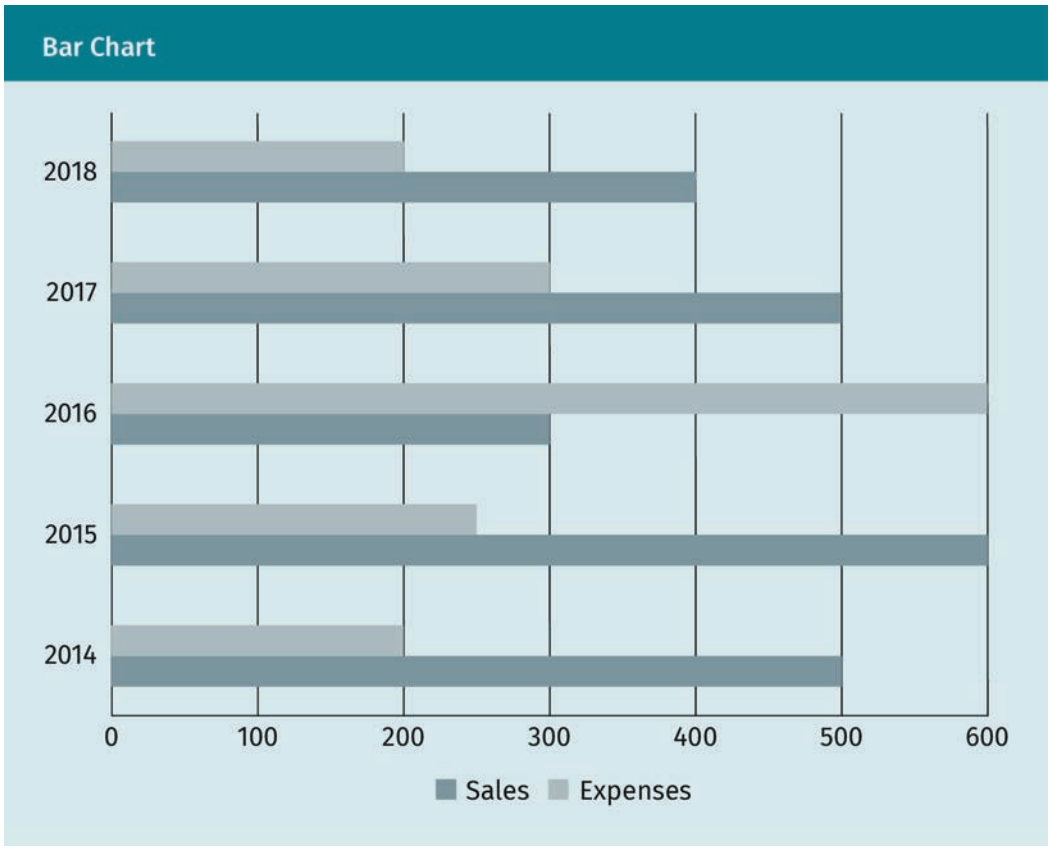
Area Charts

An area chart is based on a line chart where the area between the axis and the line represents quantitative variables.



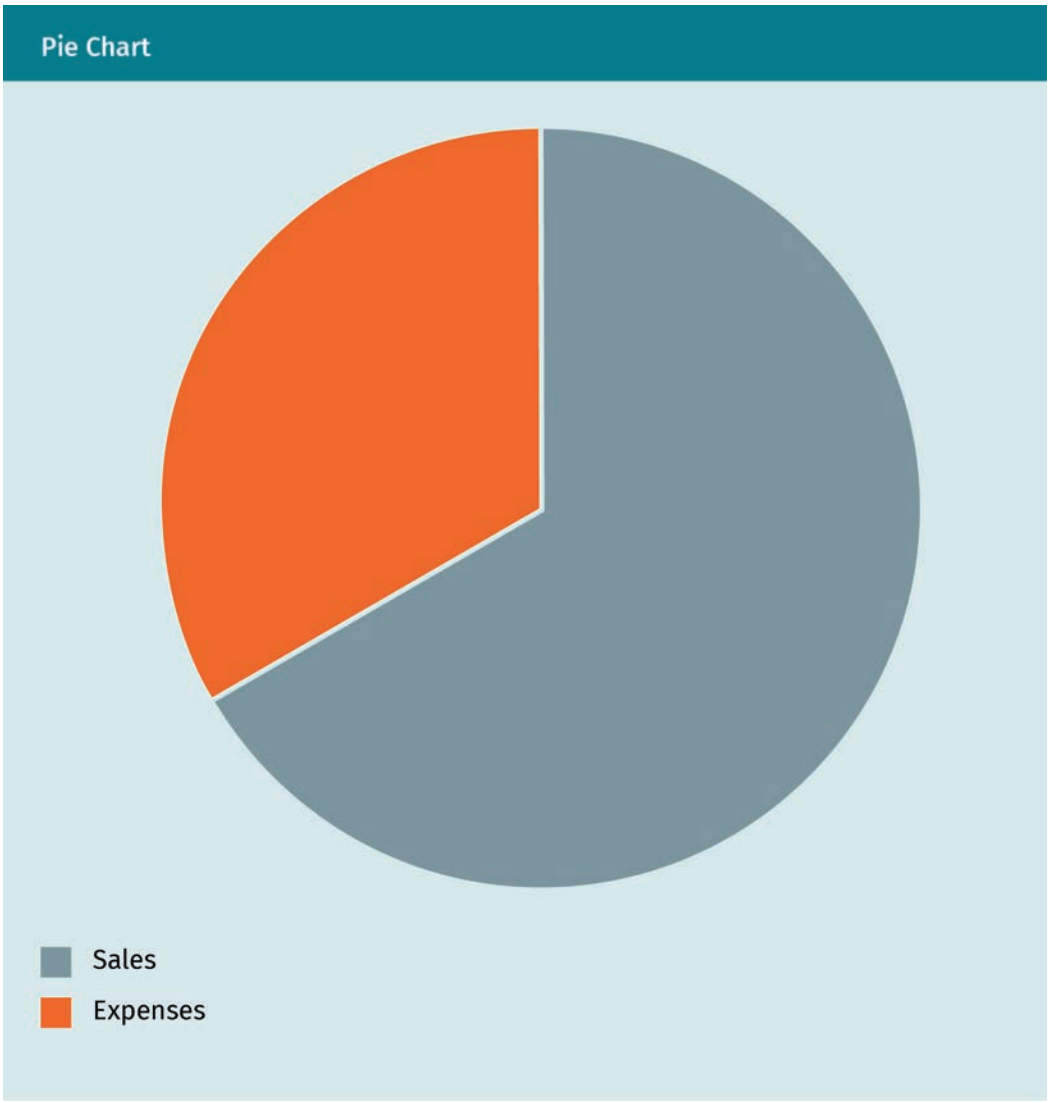
Bar Charts

A bar chart shows the variables' values in bar lengths to indicate their densities with respect to other variables.



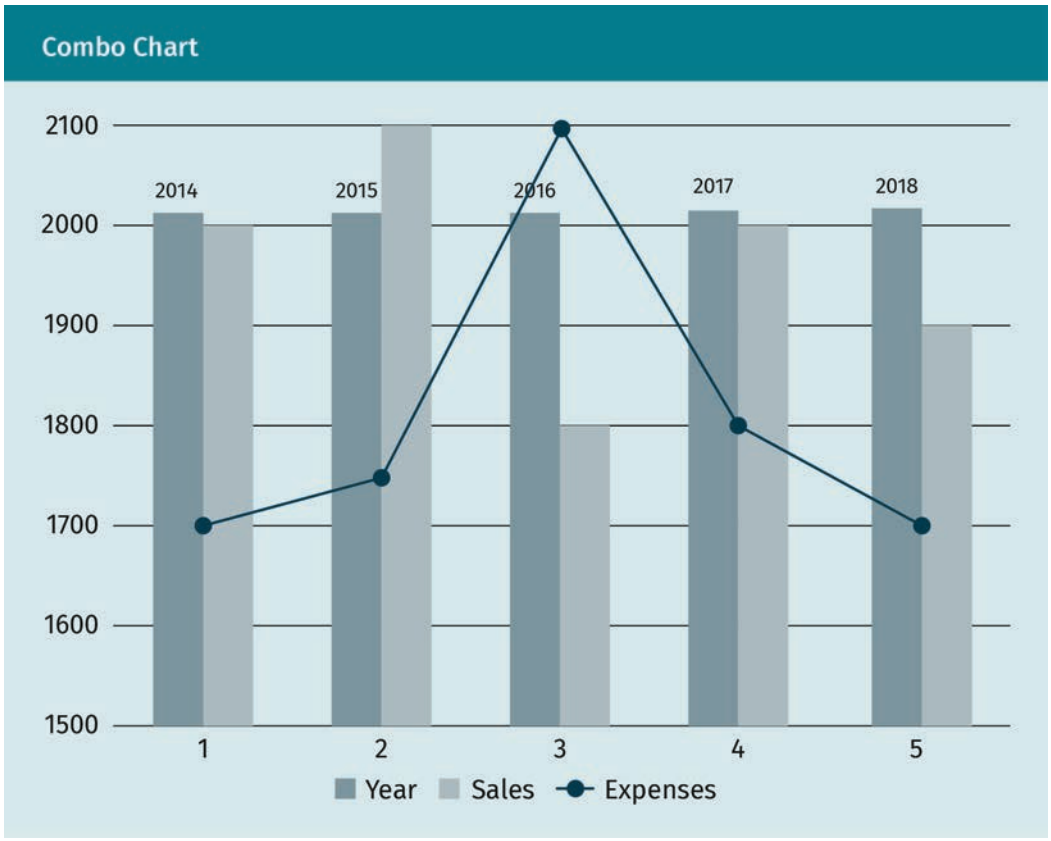
Pie Charts

The pie chart is used to show proportions of a whole, where the total of the variables values is 100 percent.



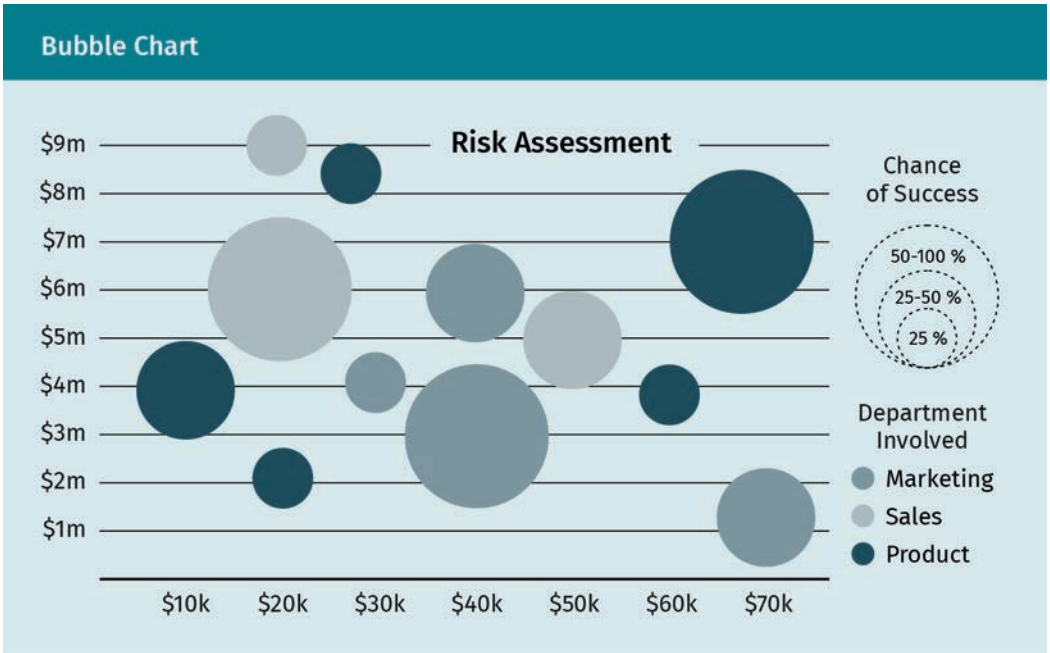
Combo Charts

The combo chart is used to highlight different types of information and is often used when the variables vary significantly.



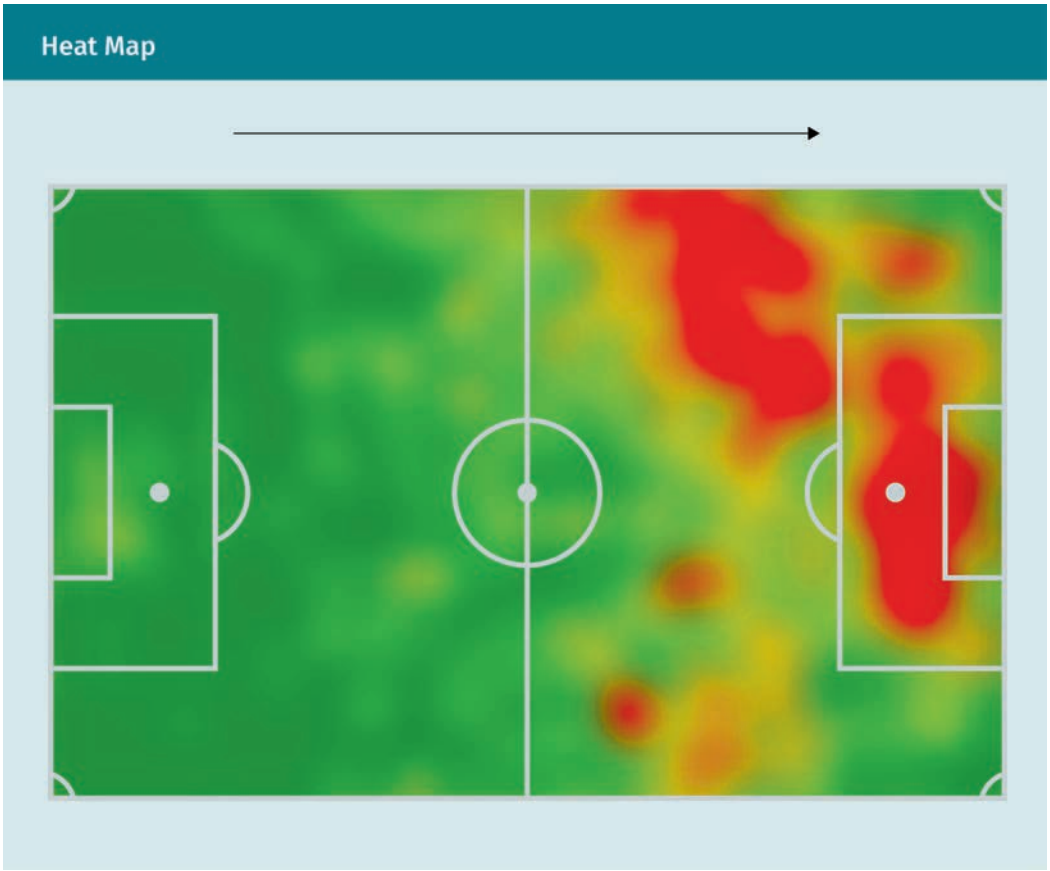
Bubble Charts

A bubble chart is used to visualize a dataset with two to four variables. The first two variables are displayed as axes values, while the third and the fourth variables are displayed as color and size, respectively.



Heat Maps

A heat map represents the values of variables in terms of a color scale to represent densities within a selected geographical area. The figure below shows the heat map for the places where one football player was most active throughout an entire game.



Summary

One of the most important steps in data science is preprocessing. In this step, the raw data are cleaned from noises and errors. The missing values and outliers are handled either by removing them entirely or estimating reasonable values for them. Duplicate records are checked and then deleted to reduce the dataset size. Furthermore, a correlation analysis is applied to avoid the presence of highly-correlated variables in the dataset.

Dataset analysis primarily requires the transformation of a dataset's variables into more representative forms. The three transformation methods are scaling, decomposition, and aggregation. With scaling, the variable is scaled to the same value range as the other variables. With decomposition, the variable is split into more than one variable to obtain a deeper overview of data variations. With aggregation, the variable is merged with one or more of the other variables for a better explanation of the dataset.

Data visualization tools are considered vital in the field of data science. The most commonly-used visualization tools are geomaps, area charts, bar charts, pie charts, combo charts, bubble charts, and heat maps.

Knowledge Check

Did you understand this unit?

Now you have the chance to test what you have learned on our Learning Platform.

Good luck!

Unit 4

Processing of Data

STUDY GOALS

On completion of this unit, you will have learned ...

... the concepts of data, information, and data processing.

... the stages and cycles of data processing.

... the different methods and types of data processing.

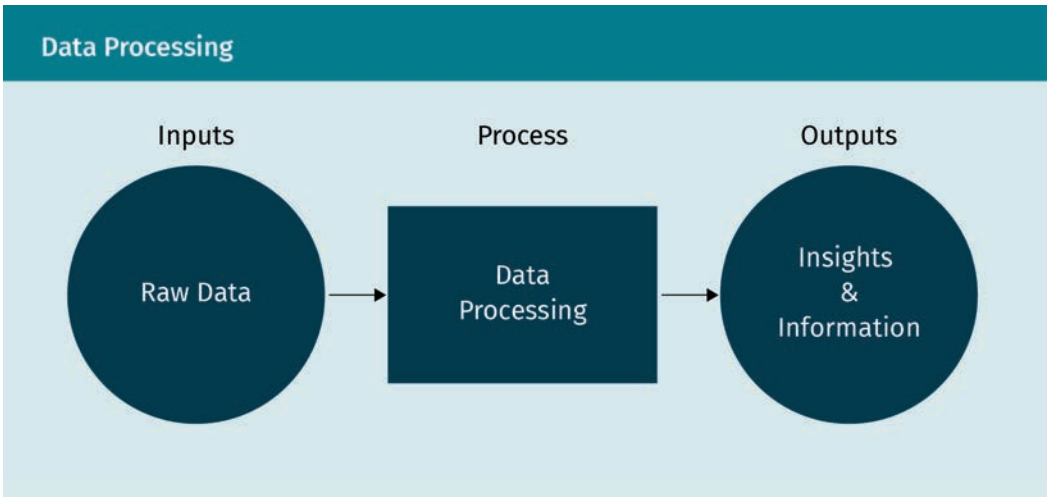
... the output forms and file formats for processed data.

4. Processing of Data

Introduction

When you buy a product from an online store, a number of data items are collected, such as your name, address, number of items bought, and amount paid. Combined they represent information about a transaction. Information transforms data into a meaningful and useful form, and information should be reliable, relevant, complete, concise, understandable, presentable, and error-free (Runkler, 2012). Organizations worldwide use this information to gain "access" to insights about sales, marketing strategies, and consumer needs. The link to this access is data processing.

Data processing is the extraction of useful information from collected raw data. It is similar to an industrial production process: inputs (raw data) are put through a particular process (data processing) to produce outputs (insights and information).



Data processing can be applied in many different scenarios such as automating office environments; administrating event tickets and reservations; managing work time and monitoring billable hours; organizing and planning the allocation of human or material resources; and conducting forecasting and optimization in an enterprise environment.

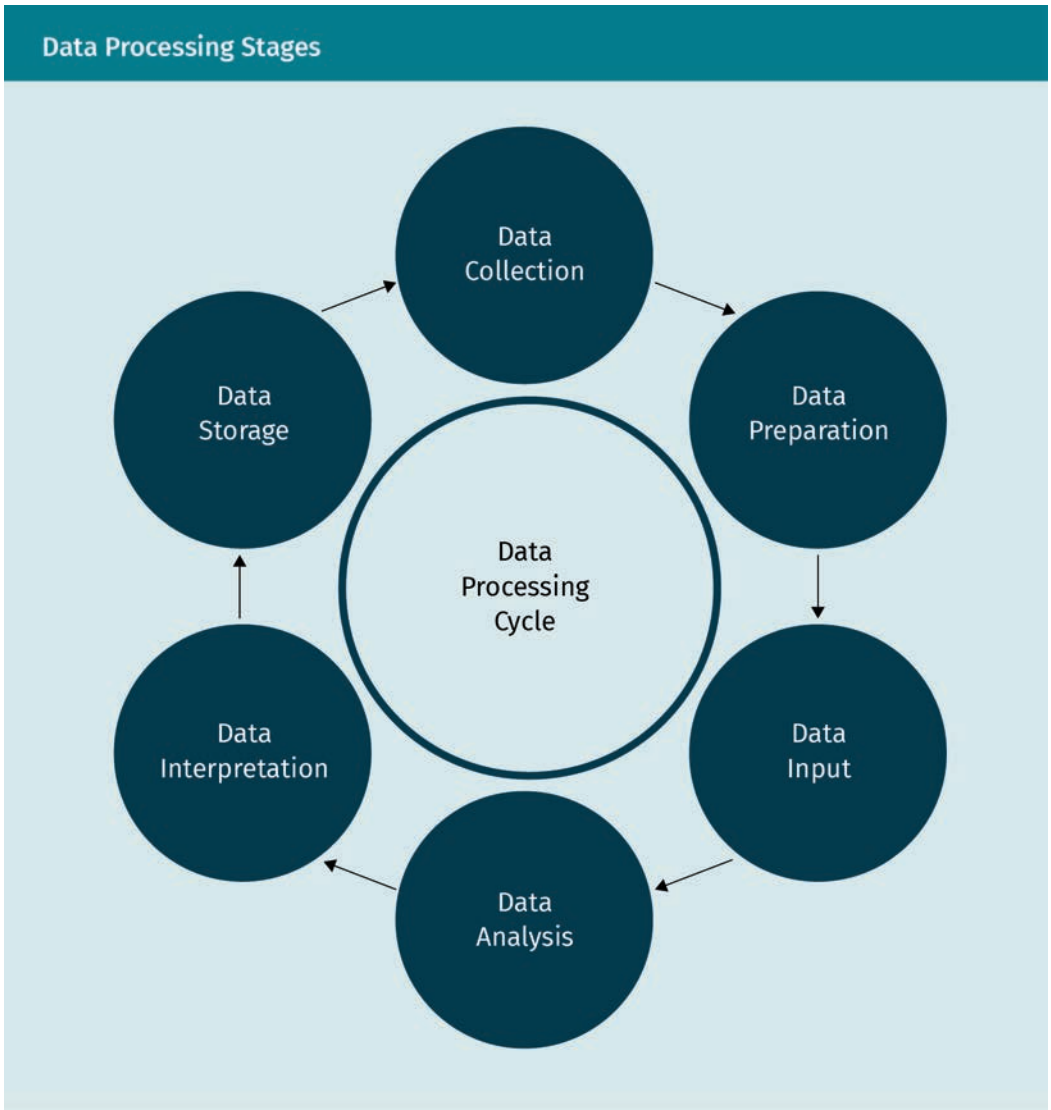
The benefits of data processing, especially in medium and large organizations, are:

- better analysis and presentation of the organization's data;
- reduction of data to only the most meaningful information;
- easier storage and distribution of data;
- simplified report creation;
- improved productivity and increased profits; and
- more accurate decision-making.

The algorithms and calculations used in data processing must be highly accurate, well-built, and correctly performed so that there is no negative effect on the decisions made based on the results.

4.1 Stages of Data Processing

Data processing stages consist of those activities necessary to transform data into information. These stages are: data collection, data preparation, input, data analysis, data interpretation, and data storage (PeerXP, 2017). Although the stages should be completed in order, this order can be considered cyclical. Often completion of the interpretation and storage stages leads to a repetition of the data collection stage.



Data Collection

Data lake
A data lake is a repository of data stored in both its natural and transformed formats.

After raw data are collected from a source(s), they are converted into a computer-friendly format (e.g., tables, text, images) to form a **data lake**. Major types of data collection include statistical populations, research experiments, sample surveys, and byproduct operations. The collection and handling of data is not always an easy task, particularly if there is noise, redundancy, and/or contradiction in the data.

Data Preparation

The data preparation stage involves preprocessing. Raw data are cleaned, organized, standardized, and checked for errors. The purpose of this stage is to deal with missing values and eliminate redundant, incomplete, duplicate, and incorrect records. Significant domain knowledge may be required to correctly prepare the data, and possession of this knowledge is important because data that are not carefully prepared and screened can result in misleading information.

Data Input

Data warehouse
A data warehouse is a store gathered from data sources and used to guide decision-making in an organization.

After the data have been prepared and cleaned, they are entered into their destination location (e.g., a **data warehouse**) and translated into a format that consumers of the data—e.g., an organization's employees—can easily understand. Understanding data means having a grasp of their key characteristics, including distribution, trends, and attribute relationships. This time-consuming process must be performed with speed and accuracy, and many organizations prefer to outsource this stage.

Data Analysis

The data analysis stage may be performed through multiple threads of simultaneously-executed instructions using machine learning and artificial intelligence algorithms. The time needed for this stage depends on the specifications of the processing device used and the complexity and amount of input data. This stage is the "heart" of data processing and may include converting the data to a more suitable format; ensuring the correctness of the data; distilling detailed data down to the main points; and combining multiple groups of data records.

The data analysis stage generally involves five steps.

Features extraction
Data are represented by a number of fixed **features** which can be categorical, binary, or continuous.

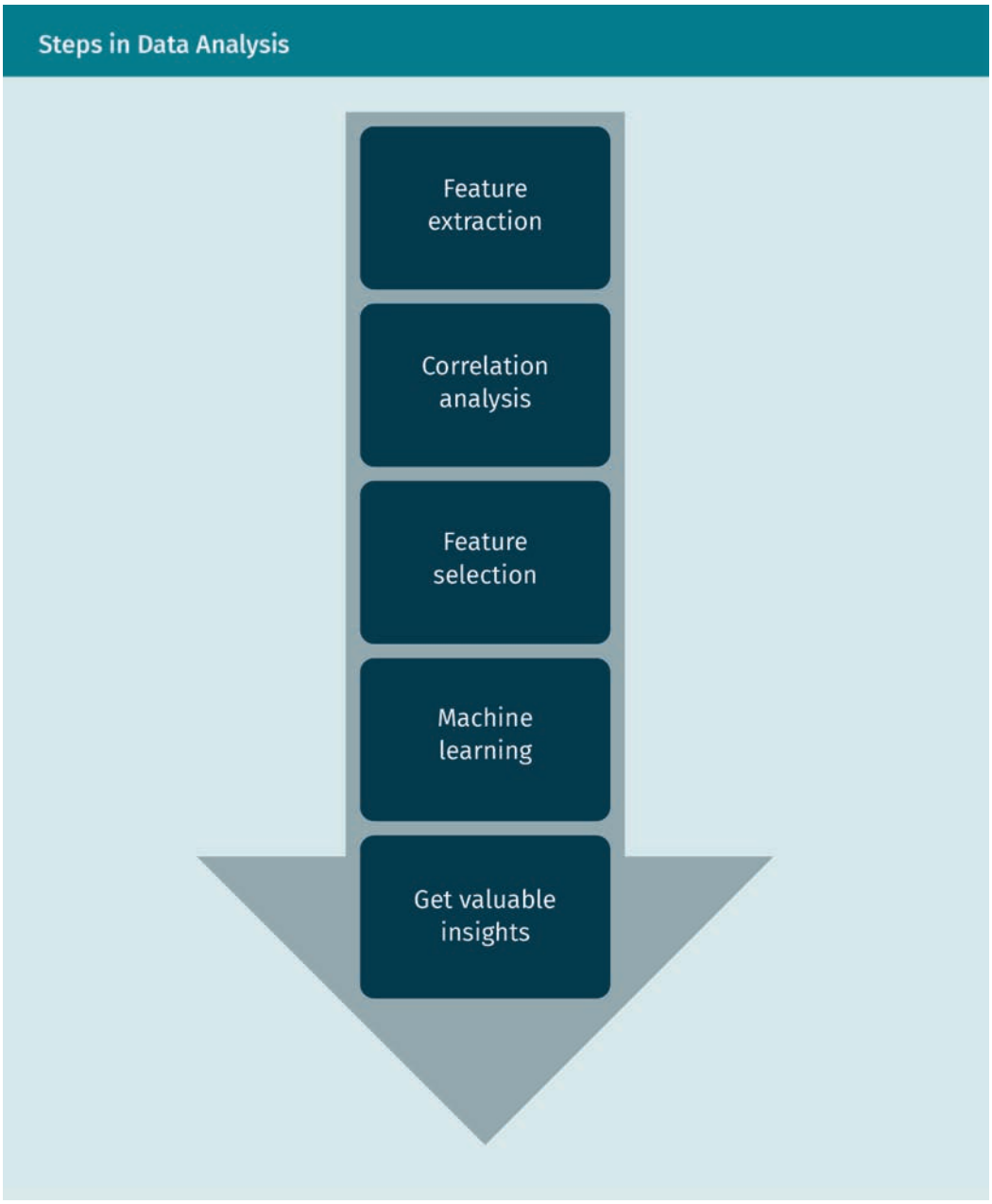
Correlation analysis
The focus of this step is to determine which pairs of data features have the highest degree of correlation. When two features are found to have a high correlation coefficient within a defined threshold, one of them can be removed from the feature set.

Data feature
A data feature, also called a variable, are aspects of the data like name, date, age, etc.

Feature selection
During this step, informative and relevant features are selected by applying correlation analysis to separate redundant features, keeping the features which show high correlation with the target variable. The result is a reduction in feature sets and data that are more comprehendable. Relevant features are those that have a low degree of intercorrelation with other features and a high level of changeability across data records. A domain expert may be needed to guide the process and review the list of suggested relevant features.

Machine learning
In this step, a learning mathematical algorithm is developed to extract knowledge from and uncover the properties of the data and predict future outcomes should new data be inserted. Descriptive analytics are used to understand underlying data patterns; predictive analytics are used to estimate new or future data based on performance; and prescriptive analytics are used to optimize the dependent action. Which learning technique to use is also determined: unsupervised (i.e., learning by applying cluster analysis) or supervised (i.e., learning by applying classification and/or regression approaches).

Extracting valuable insights
After the model is evaluated for accuracy and performance, the most important and relevant information contained in the input data is retrieved and presented. At this point the model is ready to be used for predicting future events and the probable gains/losses that a business can expect under different scenarios. However, in practice it is often difficult to judge the impact the model has on a business from the model's performance; hence, both model evaluation metrics and key performance indicators (KPIs) are used to judge the model's performance and degree of impact.



Data Interpretation

After the data analysis, it is time to interpret the data. To do so, the outcomes of the machine learning predictions need to be translated into actions. The outcomes must be interpreted to obtain beneficial information that can guide a company's future decisions. It is a critical step because the outputs of the developed model (or the model itself) need to be presented to business managers in a user-friendly form so that the managers can take appropriate actions and make better decisions. Examples of such

forms are tables, audio, videos, and images. Although the insights obtained in the data analysis stage are important, the actions taken—either automatically or as decided by humans—are the more valuable outputs.

Data Storage

The final stage of data processing is the storing the data, instructions, developed numerical models, and information for future use. Data should be stored in such a manner that they can be accessed quickly and are available for retrieval when needed.

4.2 Methods and Types of Data Processing

The methods of data processing are categorized as manual, mechanical, or electronic.

Manual Data Processing

Considered a "primitive" method, manual data processing methodology was used at a time when the technology was in its early stages and often unaffordable. It still may need to be used today for legacy data which are not digitized (e.g., historical records, maintenance logs, and patient data), and therefore all data-based calculations, transformations, and logical operations must be performed manually. Since this method of data processing is very slow, it is typically employed only by small businesses and low-capacity governmental offices. In bigger companies, the manual data processing method is avoided due to the significant time consumption and increased probability of error.

Mechanical Data Processing

In mechanical processing, data are processed using various devices like printers, calculators, and typewriters. This method is faster and more reliable than the manual data processing method but is still considered primitive.

Electronic Data Processing

With electronic processing, data are processed automatically using computer applications, software, and programs developed according to a predefined set of rules. Electronic data processing is fast and accurate. Examples include the processing of customers' bank accounts and students' university grades.

The types of the electronic data processing are batch; online; real-time; distributed (multi-processing); and time-sharing (Prakash, 2018).

Types of Electronic Data Processing	
Data processing type	Description
Batch	Input data and/or output information are grouped into batches to permit sequential processing. The tasks from different users are processed (mainly offline) in the order received.
Online	This method utilizes internet connections and attached resources (e.g., data centers, high performance computing equipment). An example of this technique is cloud computing, where data and software applications are stored in one place and employed via connection protocols in a different place.
Real-time	This approach responds almost immediately to changes in inputs and the requests of outputs. Therefore, it differs from online data processing which pays less attention to the time parameter. While real-time processing involves upfront costs for high-capacity power, the time savings are desirable because outputs are obtained in real-time. An example is banking transactions.
Distributed (multi-processing)	This method is utilized by remote workstations connected to a large server. An example is the automated teller machine (ATM), where all the back-end machines run on a certain software placed on a server, and the same information and sets of instructions are used.
Time-sharing	A single computing unit is utilized by multiple users according to a predetermined, allocated time slot. The processing is usually performed by super and mainframe computers on bulk data, such as census surveys, industry statistics, and enterprise resource planning.

4.3 Output Formats of Processed Data

Processed data should be presented in a format which meets the following criteria:

- Data files are in sophisticated formats that computers can analyze.
- People can easily recognize the data fields and their range of values.

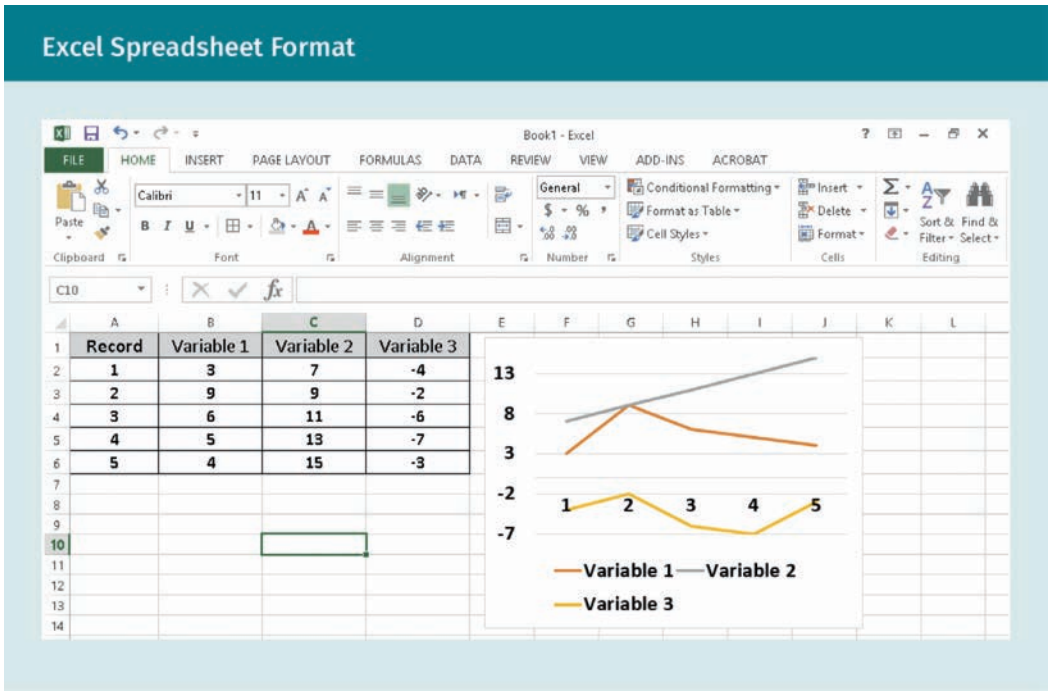
- The formats are popular and/or standard so that the data can be mixed and matched with other data resources.
- The data are clear and express the information they contain without unnecessary features (e.g., highly-correlated, redundant).

Processed data can be obtained in different forms, including:

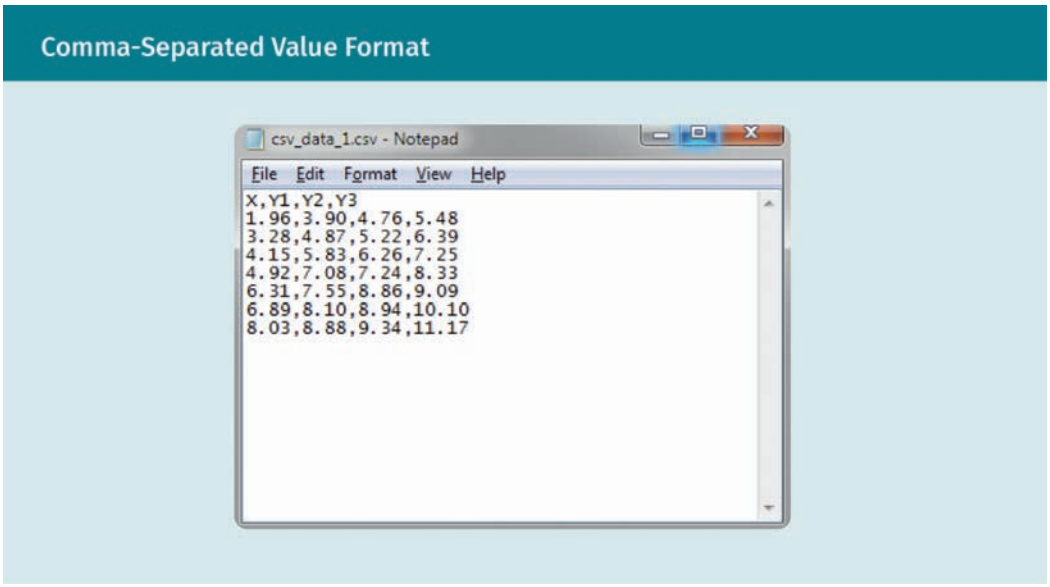
- user-readable plain text files, exported as Notepad files;
- charts to reflect trends and progress/decay;
- maps for spatial data;
- images for graphical data; and
- software-specific formats for those data requiring further analysis and processing.

There are several common software-specific data formats.

XLS (Excel spreadsheet)
This format was created by Microsoft® to be used with its Microsoft Excel product. The XLS file includes worksheets and stores data in tables consisting of rows (for records) and columns (for variables). It is also possible to create charts from the data for visualization.



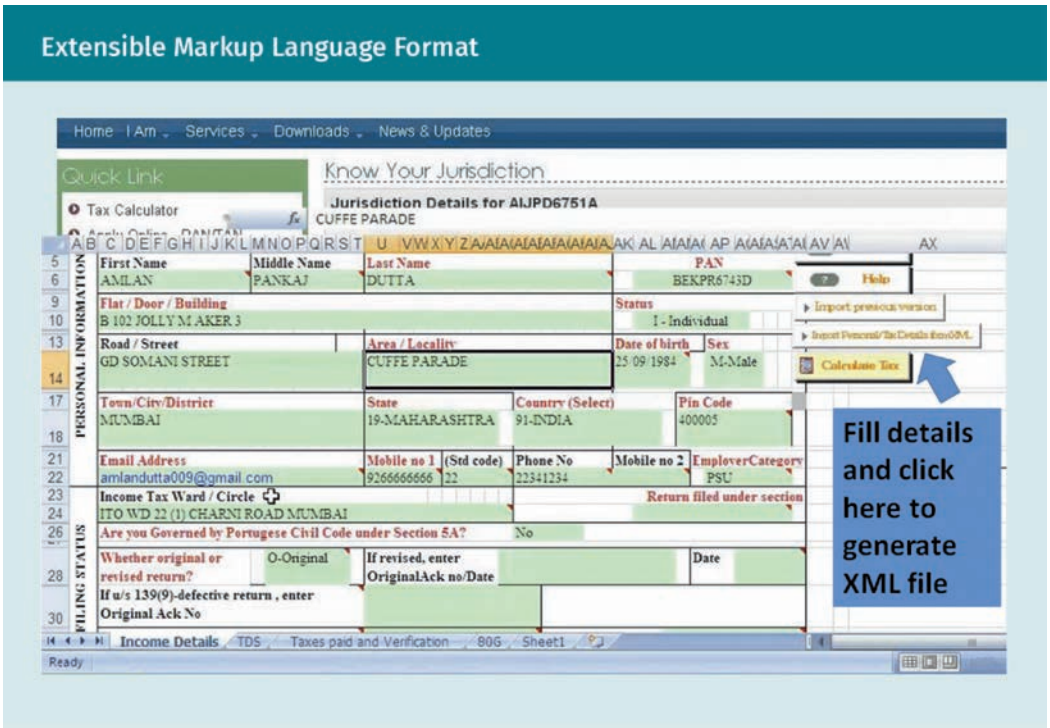
CSV (comma-separated value)
Each line in a CSV file denotes a single data record, with values separated by commas to specify the value of each data feature.



XML (extensible markup language)

This file contains structured, non-tabular data written as text with annotations. Information in an XML file is commonly formatted so that the data can be shared on the World Wide Web using **ASCII text**. It includes markup symbols to explain its contents. For example, indicates that the names of the variables are "fig" and "tag" and their values are "Alice.jpg" and "Alice", respectively. The XML file may be combined with other xml files with similar data markup symbols.

ASCII text
Abbreviation for American Standard Code for Information Interchange. ASCII code represents text for electronic communication in computers.



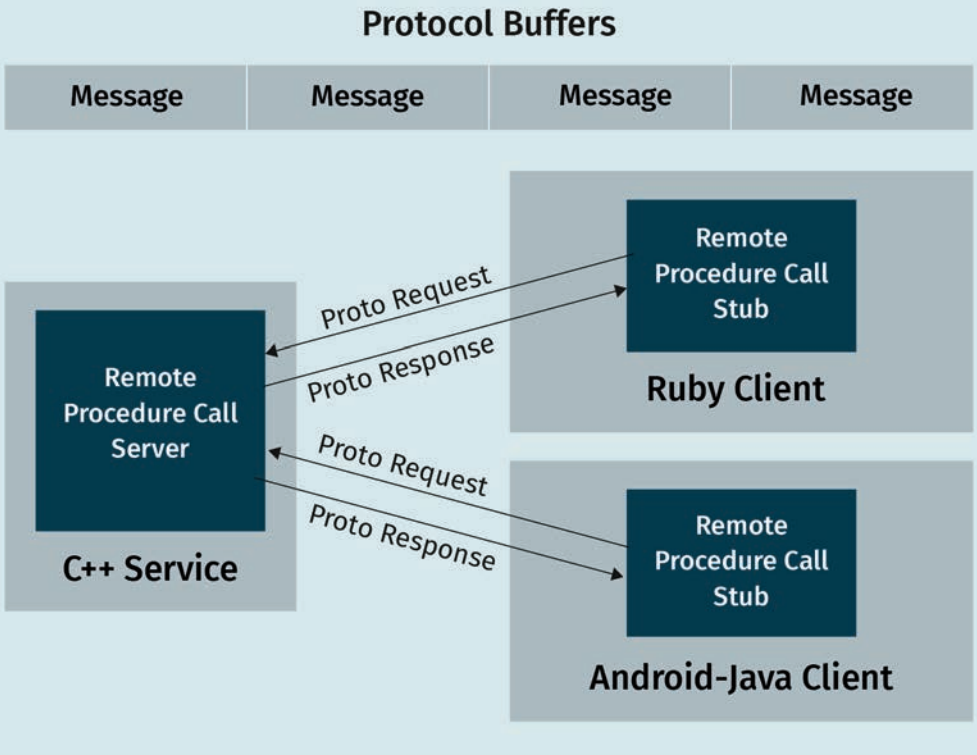
JSON (JavaScript Object Notation)
This file includes a list of variable-field pairs and their corresponding names and values. It is for the transmission of data records with their complete information among several operating systems.

Java Script Object Notation Format

```
{"employees":[
    {"firstName":"Anna","lastName":"Geier"},
    {"firstName":"Gerard","lastName":"Jones"},
    {"firstName":"Peter","lastName":"Schmidt"}
]}
```

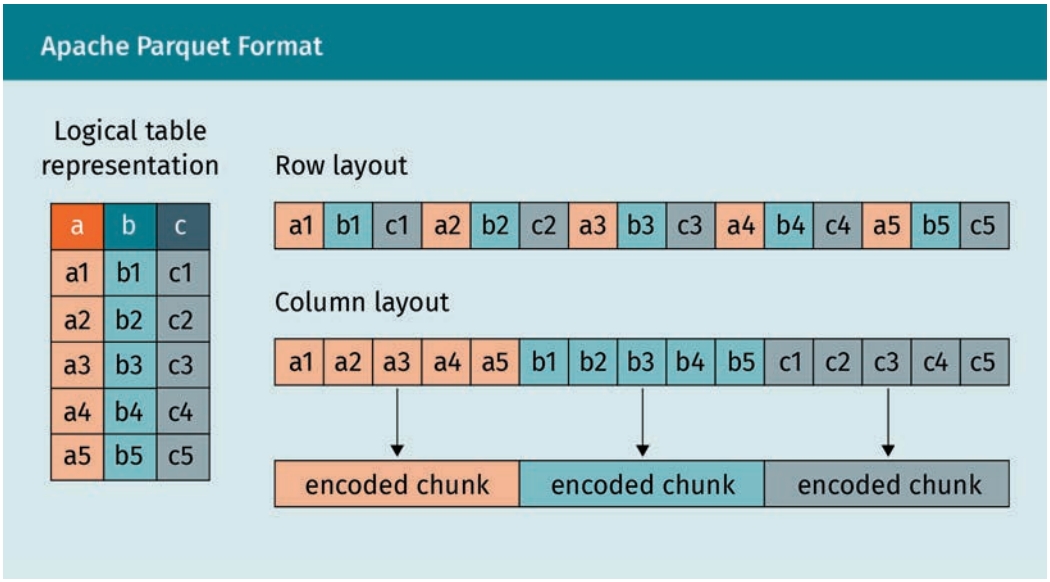
Protobuf (protocol buffers)
A reduced version of XML, these files transfer small structured data sizes across programs. This format is used for inter-application communication at Google.

Protocol Buffers Format



Apache Parquet

Apache Parquet is a column-oriented database management system format available in the Hadoop ecosystem for big data processing, regardless of the data model or programming language. Hence, it brings the ease of the traditional CSV files (which is a row based format) to the modern era with big data and more complex relations between the variables. By storing data in columns rather than rows, the database can more precisely access the data needed to answer a query, rather than scanning and discarding unwanted data in rows, as seen in the following figure. Therefore, if we query certain columns of the table, then the Apache Parquet columnar storage format is more efficient because it will read the required columns only since they are adjacent, which minimizes the inputs/outputs load. Moreover, if the data is of the same type, its storage in this column oriented format results in better data compression. Apache Parquet can also store nested structures data in columnar style, where specific fields can be accessed individually without reading all the included nested fields.

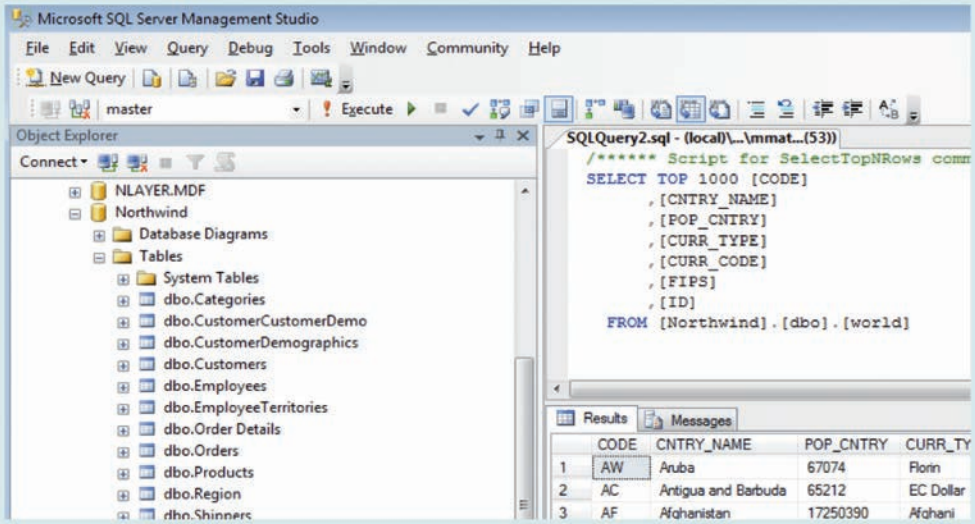


There are many other data formats which may be utilized, such as the hierarchical data format (HDF4 and HDF versions).

SQL (structured query language)

Although SQL is not a file format like the technologies discussed above, it is a widely used language for querying databases. The SQL language is descriptive in the sense that you specify what you want to have as the result, letting the interpreting system figure out a way how to achieve that goal. It provides language constructs to join, select, group, and filter tabular data.

Structured Query Language Example



Summary

Without data processing, it is almost impossible to make a good decision. It is difficult to think of any industry that does not implement data processing to obtain insights into areas which require improvements.

Data processing is a multidimensional process that starts with the collection of an immeasurable amount of data from various sources. The data are arranged in practical, organized forms and forwarded to the next stage, data preparation. In this stage, all preprocessing operations are performed on the data to remove noise and outliers. The data are then entered into the computer in a usable form. The data analysis stage converts the raw data into meaningful insights and information by using a machine learning model. Machine learning is employed to perform a series of operations on the preprocessed data so that relations within the data elements can be presented in various forms such as distribution curves, reports, and images.

Data processing can be manual, mechanical, or electronic. The latter is the fastest and most accurate method and includes many processing types: batch, online, real-time, distributed (multi-processing), and time-sharing. The common formats of a data processed file are XLS, CSV, XML, SQL, JSON, and Protobuf. In each type of file, the data are presented in a predetermined structure specified by the properties of the associated formats.

Knowledge Check

Did you understand this unit?

Now you have the chance to test what you have learned on our Learning Platform.

Good luck!

Unit 5

Selected Mathematical Techniques

STUDY GOALS

On completion of this unit, you will have learned ...

- ... how to apply principal component analysis to data.
- ... how to perform cluster analysis on a dataset.
- ... how to describe the linear regression model and compute its coefficients.
- ... how to describe the important features of time-series data.
- ... the popular models for forecasting future values in time-series data.
- ... the common approaches for dataset transformation.

5. Selected Mathematical Techniques

Introduction

In this unit, the mathematical techniques and models used to transform data into insightful information will be discussed. These models are employed to cluster the input data and/or predict the performance of the input data in new scenarios. There are two modeling approaches for prediction: regression and classification. The aim of regression is to predict a numerical value for a variable (e.g., forecasting an organization's future revenue after three years). The aim of classification is to predict the best category for a variable (e.g., forecasting tomorrow's weather as sunny, cloudy, or rainy). In general, the best model to represent the data is not necessarily the one with the highest training accuracy, but the one with the best sense of the data's boundaries and limitations.

We begin this unit with a discussion of principal component analysis, a common technique for dimensionality reduction of the input data to their relevant variables. A detailed overview will be given of data clustering analysis as a tool for unsupervised learning and the grouping of data records into unlabeled clusters according to the level of their characteristic similarity. The linear regression technique for predicting linear varied data behavior will also be presented.

Because most predictive analysis techniques are applied to datasets with time as the main variable, we have included a separate section on designing forecasting models for time-series data. It may be necessary to transform the data to another domain for better or further understanding and analysis before developing the cluster and/or regression models (e.g., dataset is given in the time domain, but it is more understandable in the frequency domain). Therefore, basic transformation techniques are also discussed in this unit.

5.1 Principal Component Analysis

Input data usually include a considerable number of correlated (e.g., redundant and/or irrelevant) variables which place a burden on prediction models. If the correlation between two variables is not 100 percent (and in most cases, it is not), it means there is some amount of independent information contained within each variable. However, the complexity of conducting data analysis with a long list of variables is too high. Therefore, a threshold might need to be set for an acceptable correlation percentage (e.g., 80 percent). When a pair of variables reaches this percentage, the variables are considered correlated; therefore, one of them is assumed to be redundant and can be removed safely from the dataset.

Principal component analysis (PCA) is applied to transform linearly-correlated variables into uncorrelated variables called principal components (PCs). PCA also sorts the produced uncorrelated variables according to their variance along the data records. A

decision can then be made to exclude the variables appearing at the bottom of the PC list (i.e., those with low changeability), resulting in a desired reduction in dimensionality of the dataset.

Principal component analysis
This is a statistical analysis method applied in order to transform potentially correlated variables into uncorrelated variables (principle components).

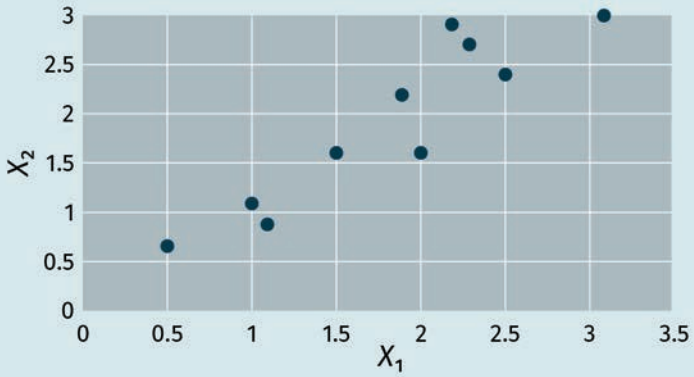
The aim is to construct a new set of variables from the old, such that most information is contained within the first few variables. This makes it easier to define a cutoff and possible to use only a subset of the new variables in the next step (e.g., a machine learning model or regression).

The first PC accounts for much of the variability in the data. PCA initially seeks linear combinations of the original variables weighted by their contribution, in order to extract the maximum variance. This variance is then removed, and PCA seeks the second linear combination of these variables that explains the second maximum variance, forming the second PC. Each succeeding PC accounts for much of the remaining data variability.

The figure shows a dataset with two variables, x_1 and x_2. The first principal component (PC_1) is the axis along which the data records show the largest variation. The second principal component (PC_2) is the axis along the second highest variation and orthogonal to (PC_1).

PCA Example

#	x_1	x_2
1	2.5	2.4
2	0.5	0.7
3	2.2	2.9
4	1.9	2.2
5	3.1	3
6	2.3	2.7
7	2	1.6
8	1	1.1
9	1.5	1.6
10	1.1	0.9



PCA Algorithm

Step (1): Get and subtract the mean

For an input dataset with N records (1, 2, ..., N) and M variables (x_1, x_2, ..., x_M), the mean of each variable is calculated using the following equation:

$$\overline{x}_i = \frac{1}{N}\sum_{k=1}^{N} x_{i_k}$$

$$i = 1, 2, ..., M$$

The calculated mean is then subtracted from its associated variable for all data records by the equation given below. This step produces a dataset with a mean of zero and simplifies the remaining steps of the PCA algorithm.

$$x_i = x_i - \overline{x}_i$$
$$i = 1, 2, ..., M$$

Step (2): Calculate the covariance matrix

The covariance $C(x_i, x_j)$ is a measure of the changes in variable x_i with respect to changes in variable x_j, according to the following equation:

$$C(x_i, x_j) = \frac{1}{N-1}\sum_{k=1}^{N} (x_i \cdot x_j)_k$$

Since the covariance is calculated for all data variables with respect to each other, this will form a symmetric matrix with dimensions $[M \cdot M]$, as shown below.

$$C = \begin{bmatrix} C(x_1, x_1) & C(x_1, x_2) & C(x_1, x_3) & \cdots & C(x_1, x_M) \\ C(x_2, x_1) & C(x_2, x_2) & C(x_2, x_3) & \cdots & C(x_2, x_M) \\ C(x_3, x_1) & C(x_3, x_2) & C(x_3, x_3) & \cdots & C(x_3, x_M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C(x_M, x_1) & C(x_M, x_2) & C(x_M, x_3) & \cdots & C(x_M, x_M) \end{bmatrix}$$

The exact value of $C(x_i, x_j)$ is an indication of how strongly the two variables depend on each other; however, the value is not as important as the sign. A positive covariance indicates that both variables increase (or decrease) together, while a negative value indicates that if one variable increases, the other variable decreases (or vice versa). If the covariance is zero, the variables are uncorrelated.

Step (3): Calculate the eigenvalues and eigenvectors

The objective of PCA is to transform the calculated covariance matrix into an optimum form where all the variables are uncorrelated linearly to first order (i.e., $C(x_i, x_j) = 0$, $i \neq j$). This results in a diagonal matrix where all elements equal zero except those in the diagonal, as presented by the following equation:

$$C = \begin{bmatrix} C(x_1, x_1) & C(x_1, x_2) & C(x_1, x_3) & \cdots & C(x_1, x_M) \\ C(x_2, x_1) & C(x_2, x_2) & C(x_2, x_3) & \cdots & C(x_2, x_M) \\ C(x_3, x_1) & C(x_3, x_2) & C(x_3, x_3) & \cdots & C(x_3, x_M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C(x_M, x_1) & C(x_M, x_2) & C(x_M, x_3) & \cdots & C(x_M, x_M) \end{bmatrix} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_M \end{bmatrix}$$

$$\therefore C - \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_M \end{bmatrix} = 0$$

$$C - [\lambda_1 ... \lambda_M] \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} = 0$$

$$C - [\lambda_1 ... \lambda_M] \cdot I = 0$$

where the diagonal elements of the transformed matrix are called eigenvalues (λ), and I denotes the identity matrix (i.e., a matrix with "1" in its diagonal and "0" otherwise). The eigenvalues are found by solving the following equation:

$$\det(C - \lambda \cdot I) = |C - \lambda \cdot I| = 0$$

where $\det$ is the determinant of the matrix.

The principal components (PCs) are the Eigenvectors of the calculated eigenvalues. An eigenvector is a vector which, when transformed by the covariance matrix, results in a scaled version of the vector, and this scale is the associated eigenvalue, as explained in the following equation:

$$C \cdot PC_i = (\lambda_i \cdot I) \cdot PC_i$$
$$i = 1, 2, ..., M$$

Therefore, the solution to $[(C - \lambda_i \cdot I) \cdot PC_i = 0]$ will result in the i^{th} principal component (PC_i). Since there are no correlations between the obtained PCs, the eigenvectors are orthogonal vectors.

Step (4): Formulate the PCs

The PC (i.e., eigenvector) that corresponds to the highest eigenvalue is the first principal component of the dataset. Consequently, the next step is to order all other PCs according to their eigenvalues, from highest to lowest. The percentage of how much variance (V) each PC represents is calculated by the following equation:

$$H_{PC_i} = \frac{\lambda_i}{\lambda_1 + \cdots + \lambda_M} \cdot 100\,\%$$

Step (5): Dimensionality reduction
We may decide to ignore the PCs with less significance (i.e., those that appear at the bottom of the PC list) due to their low eigenvalues. This will reduce the original dataset of (x_1, x_2, ..., x_M) variables to a smaller version with (PC_1, PC_2, ..., PC_M), where $M^* < M$.

Step (6): Reconstruct the dataset
The dataset is now reconstructed by the produced PCs, using the following equation:

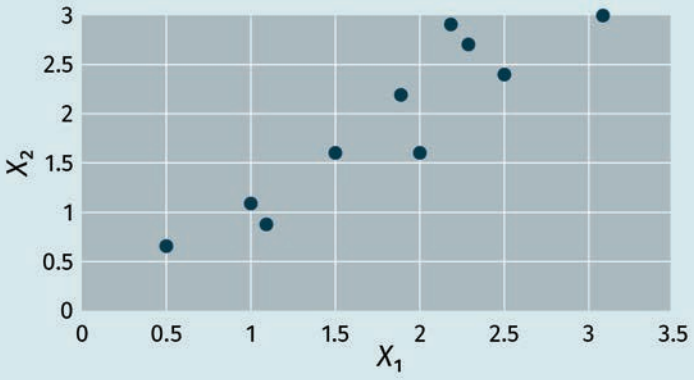
$$[y]^T = [PC_1 ... PC_{M*}]^T \cdot [x]^T$$

PCA Example

For the dataset given in the figure, follow the PCA algorithm to develop a new version of the dataset in which its variables are uncorrelated and ordered according to their significance to the input data.

PCA Example

#	x_1	x_2
1	2.5	2.4
2	0.5	0.7
3	2.2	2.9
4	1.9	2.2
5	3.1	3
6	2.3	2.7
7	2	1.6
8	1	1.1
9	1.5	1.6
10	1.1	0.9



The dataset is shown using the "normal" x_1 and x_2 axes, but looking at the distribution, this does not seem to be optimal. The data records are scattered around the diagonal ($x_1 = x_2$), so we would expect that the diagonal itself would be a better primary axis as it captures the most important variance of the data records. Since the data records are not all on the diagonal, we expect that a second axis perpendicular to the diagonal will capture the second-highest variability of these data records. Hence, the information in the graph is better described using the diagonal and a new axis perpendicular to it. If we need to reduce the number of variables, we could use only the new (diagonal) axis, as it captures most of the information, and neglect the second new axis which contains less significant information about the variance of the data points. The algorithm is performed as follows:

- The mean values are: $\overline{x}_1 = \frac{1}{10} \cdot 18.1 = 1.81$, $\overline{x}_2 = \frac{1}{10} \cdot 19.1 = 1.91$
- Subtract the mean from the dataset: $x_1 = x_1 - 1.81$, $x_2 = x_2 - 1.91$

Adjusted Dataset (Zero Mean)		
#	x_1	x_2
1	0.69	0.49
2	–1.31	–1.21
3	0.39	0.99
4	0.09	0.29
5	1.29	1.09
6	0.49	0.79
7	0.19	–0.31
8	–0.81	–0.81
9	–0.31	–0.31
10	–0.71	–1.01

- Calculate the covariance matrix:

$$C = \begin{bmatrix} C(x_1, x_1) & C(x_1, x_2) \\ C(x_2, x_1) & C(x_2, x_2) \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 5.549 & 5.539 \\ 5.539 & 6.449 \end{bmatrix} = \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

- Calculate the eigenvalues:

$$\det(C - \lambda \cdot 1) = 0$$

$$\begin{vmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{vmatrix} = 0$$

$$(0.6165-\lambda)(0.7165-\lambda) - (0.6154)^2 = 0$$

$$\left(0.4417-1.333\lambda + \lambda^2\right) - 0.3787 = 0$$

$$\lambda^2 - 1.333\lambda + 0.063 = 0$$

$$\therefore \lambda_1 = 1.284,\ \lambda_2 = 0.049$$

- Calculate the eigenvectors:
  PC_1:

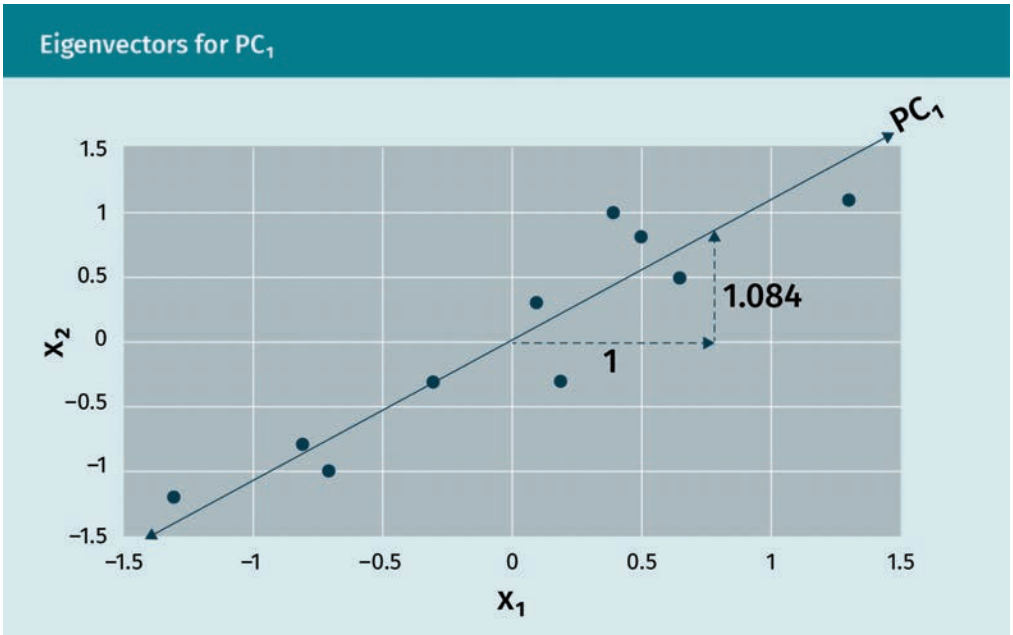
$$(C - \lambda_1 \cdot I) \cdot PC_1 = 0$$

$$\begin{pmatrix} -0.6675 & 0.6154 \\ 0.6154 & -0.5675 \end{pmatrix} \cdot \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = 0$$

$$-0.6675a_1 + 0.6154b_1 = 0$$

$$0.6154a_1 - 0.5675b_1 = 0$$

$$\therefore PC_1 = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.084 \end{bmatrix}$$



PC_2:

$$(C - \lambda_2 \cdot I) \cdot PC_2 = 0$$

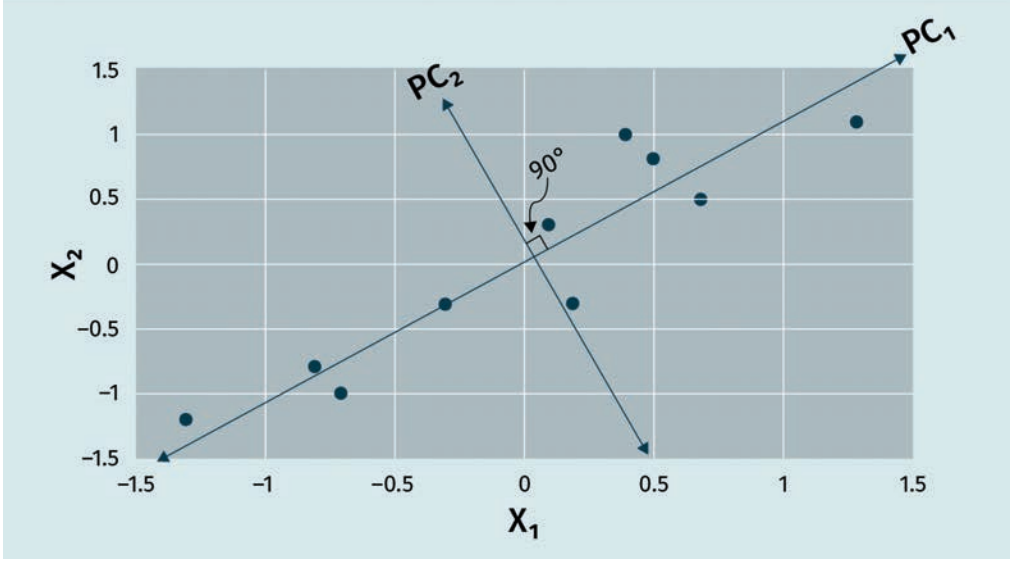
$$\begin{pmatrix} 0.5675 & 0.6154 \\ 0.6154 & 0.6675 \end{pmatrix} \cdot \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = 0$$

$$0.5675a_2 + 0.6154b_2 = 0$$

$$0.6154a_2 + 0.6675b_2 = 0$$

$$\therefore PC_2 = \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1.084 \\ 1 \end{bmatrix}$$

Eigenvectors for PC_1 and PC_2



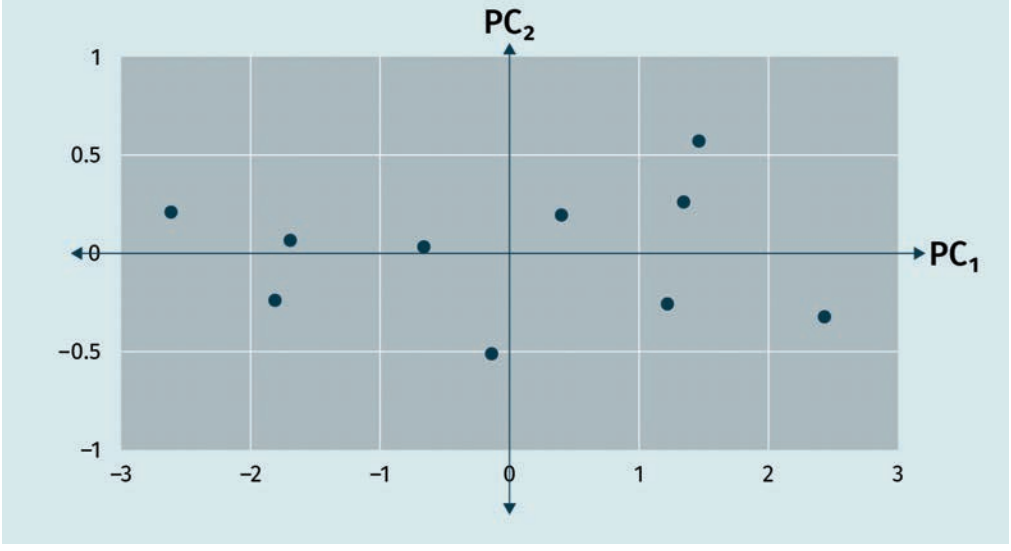
- Reconstruct the dataset:
  new data samples: $y_i = [PC_1 \quad PC_2]^T \cdot [x_i]^T$

Reconstructed Data

#	y_1	y_2
1	1.22	−0.25
2	−2.62	0.21

#	y_1	y_2
3	1.46	0.56
4	0.40	0.19
5	2.47	−0.30
6	1.34	0.25
7	−0.14	−0.51
8	−1.68	0.06
9	−0.64	0.02
10	−1.80	−0.24

Reconstructed Graph



5.2 Cluster Analysis

Clustering is an unsupervised learning technique that permits the input data to be grouped into unlabeled, meaningful clusters. Each cluster includes a group of data records which share a certain level of similarity (defined for the underlying dataset) and at the same time are dissimilar to the data records in other clusters. The number of clusters that can be formed from the input data depends on the context and the "eye of the beholder." There are many clustering approaches such as K-means, expectation maximization, agglomerative, density-based spatial, and affinity propagation. For our purposes we limit the discussion to the two main approaches, K-means clustering and agglomerative clustering.

Clustering
This is the method of grouping objects together such that objects in the same group have more in common with one another than those objects in other groups.

K-Means Clustering

K-means clustering is an algorithm for grouping a given N data records into K clusters. The algorithm is straightforward and can be executed in the following steps (Runkler, 2012):

1. Decide on the number of clusters K.
2. Select random data records to represent the centroids of these clusters.
3. Calculate the distances between each data record and the defined centroids. Then assign the data record to the cluster containing the centroid closest to that data record (i.e., centroid which returned the minimum distance). The Euclidean distance $\mathrm{d}_{(\mathrm{i},\ \mathrm{C})}$ is the distance measurement during K-means clustering, as given by the equation:

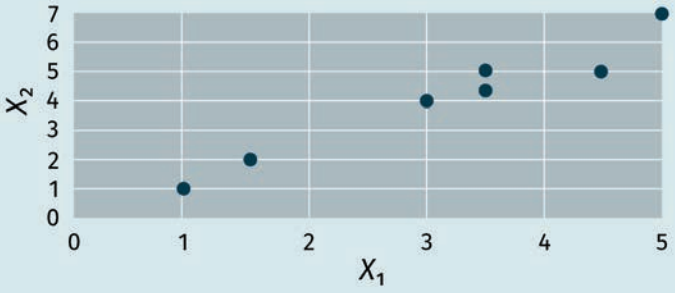
$$\mathrm{d}_{(\mathrm{i},\ \mathrm{c})} = \sqrt{\left(\mathrm{x}_{1,\ \mathrm{i}} - \mathrm{x}_{1,\ \mathrm{c}}\right)^2 + \left(\mathrm{x}_{2,\ \mathrm{i}} - \mathrm{x}_{2,\ \mathrm{c}}\right)^2 + \cdots + \left(\mathrm{x}_{\mathrm{M},\ \mathrm{i}} - \mathrm{x}_{\mathrm{M},\ \mathrm{c}}\right)^2}$$

   where $(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{M})$ are the M data variables, i denotes the i^{th} data record, and c denotes the cluster's centroid.
4. Recalculate the new centroid for each cluster by averaging the included data records.
5. Repeat steps (3) and (4) until there are no further changes in the calculated centroids.
6. The final clusters comprise the data records included within them.

Example
Apply K-means clustering on the following dataset containing two variables and seven data records.

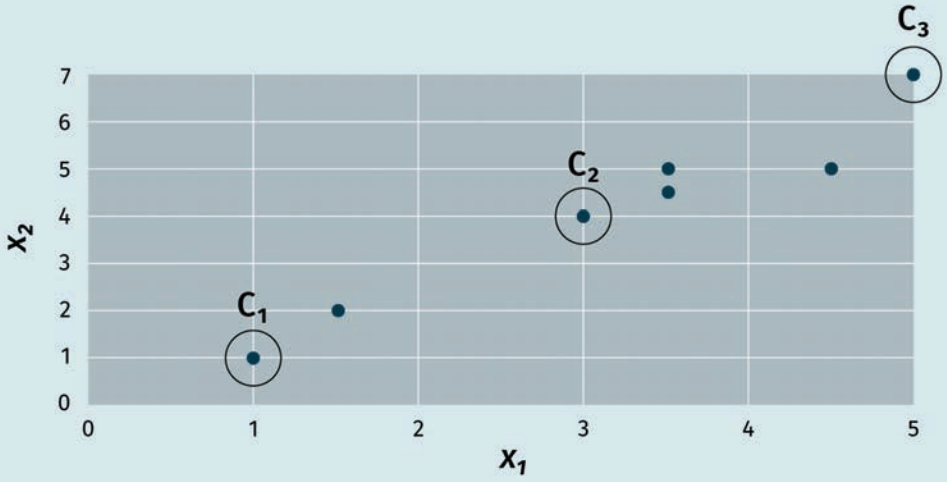
K-Means Clustering Example

#	x_1	x_2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5



1. It is assumed that there are three clusters (i.e., $\mathrm{K} = 3$): C_1, C_2 and C_3.
2. The centroids for these clusters are selected to be the first (1, 1), third (3, 4), and fourth (5, 7) data records.

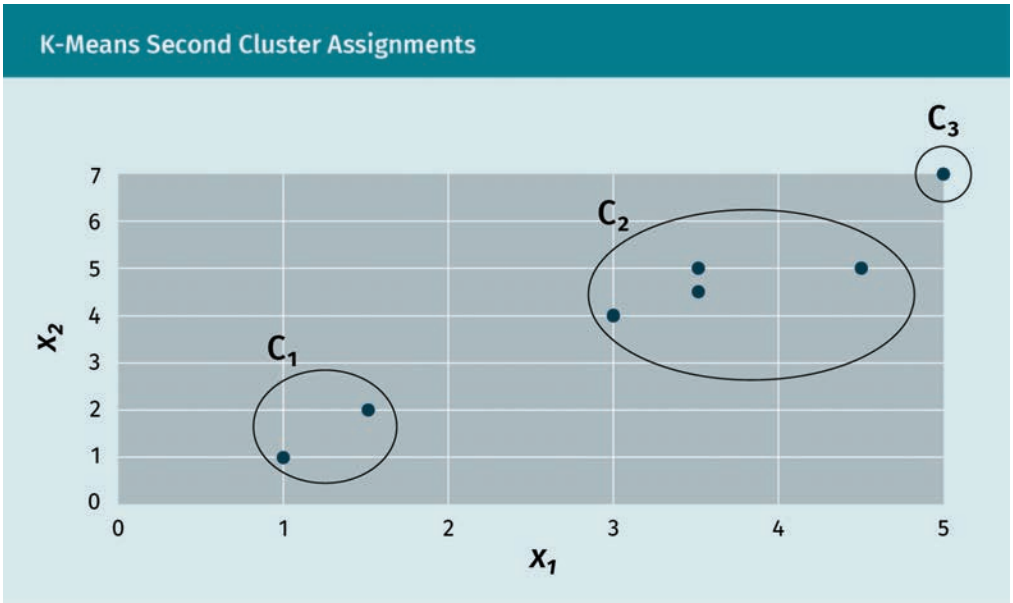
K-Means First Cluster Assignments



The distances between each data record and the defined centroids are:

$$\mathrm{d}_{(2,\,\mathrm{C}_1)} = \sqrt{(1.5-1)^2 + (2-1)^2}$$
$$\mathrm{d}_{(2,\,\mathrm{C}_2)} = \sqrt{(1.5-3)^2 + (2-4)^2}$$
$$\mathrm{d}_{(2,\,\mathrm{C}_3)} = \sqrt{(1.5-5)^2 + (2-7)^2}$$
$$\therefore \text{Record } \#2 \text{ is assigned to } \mathrm{C}_1$$

In the same manner, the other data records are assigned to their closest clusters as follows: record #5 is assigned to C_2, record #6 is assigned to C_2, and record #7 is assigned to C_2.



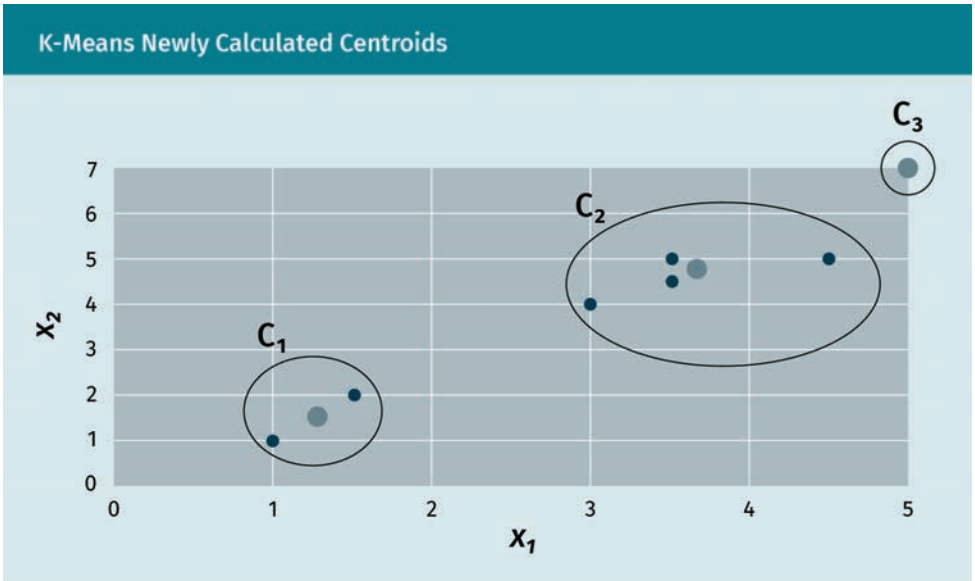
3. Recalculate the new centroid for each cluster by averaging its included data records.
   - For C_1, the centroid is:

$$\left(\frac{1}{2}(1+1.5,\ 1+2)\right) = \frac{1}{2}(2.5,\ 3) = (1.25,\ 1.5)$$

   - For C_2, the centroid is:

$$\left(\frac{1}{4}(3+3.5+3.5+4.5,\ 4+4.5+5+5)\right) = (3.625,\ 4.625)$$

   - For C_3, the centroid remains as it is at (5, 7)



4. Calculate the new distances between each data record and cluster centroids and assign data records to clusters according to their minimum distances.

New Distances Between Data Records and Cluster Centroids

#	Distance to C_1	Distance to C_2	Distance to C_3
1	**0.559017**	4.475628	7.211103
2	**0.559017**	3.377314	6.103278
3	3.051639	**0.883883**	3.605551
4	6.656763	2.744312	**0**
5	4.160829	**0.395285**	2.5
6	4.776243	**0.951972**	2.061553
7	3.75	**0.176777**	2.915476

Since there are no changes in the assignments of the data records to the clusters, the centroids remain at their previous values, and there is no need to proceed with more iterations. Therefore, the final cluster contents are: C_1: {#1, and #2}, C_2: {#3, #5, #6, and #7}, and C_3: {#4}.

Hierarchical Clustering

Hierarchical clustering is applied to data that has an underlying hierarchy. For example, consider the following items in a supermarket: apples and pears are fruit; tomatoes and cucumbers are vegetables; and both are fresh produce.

There are two approaches to hierarchical clustering: divisive (top-down) and agglomerative (bottom-up).

Agglomerative Clustering

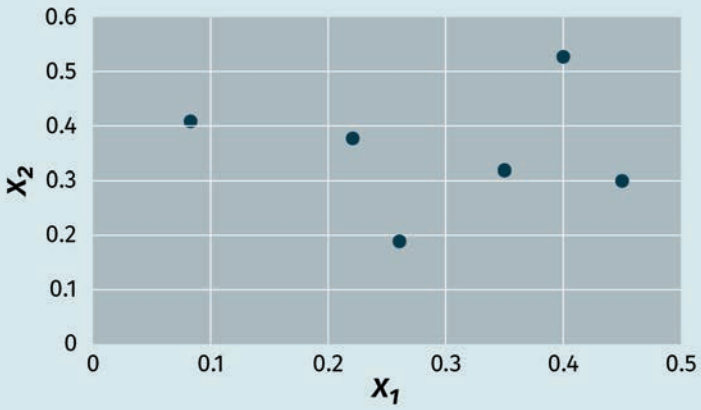
Agglomerative clustering creates a bottom-up tree (dendrogram) of clusters that repeatedly merge two nearest points or clusters into a bigger super cluster. The "leaves" of the tree are the individual data records, and the "root" is the universe of these records. The agglomerative clustering algorithm is formulated as follows:

1. Assign each record of the given N data records to a unique cluster, forming N clusters.
2. Merge the data records (i.e., clusters) with minimum Euclidean distance between them into a single cluster.
3. Repeat this process until there is only one cluster remaining, forming a hierarchy of clusters.

Example
Apply agglomerative clustering to the following dataset containing two variables and six data records.

Agglomerative Clustering Example

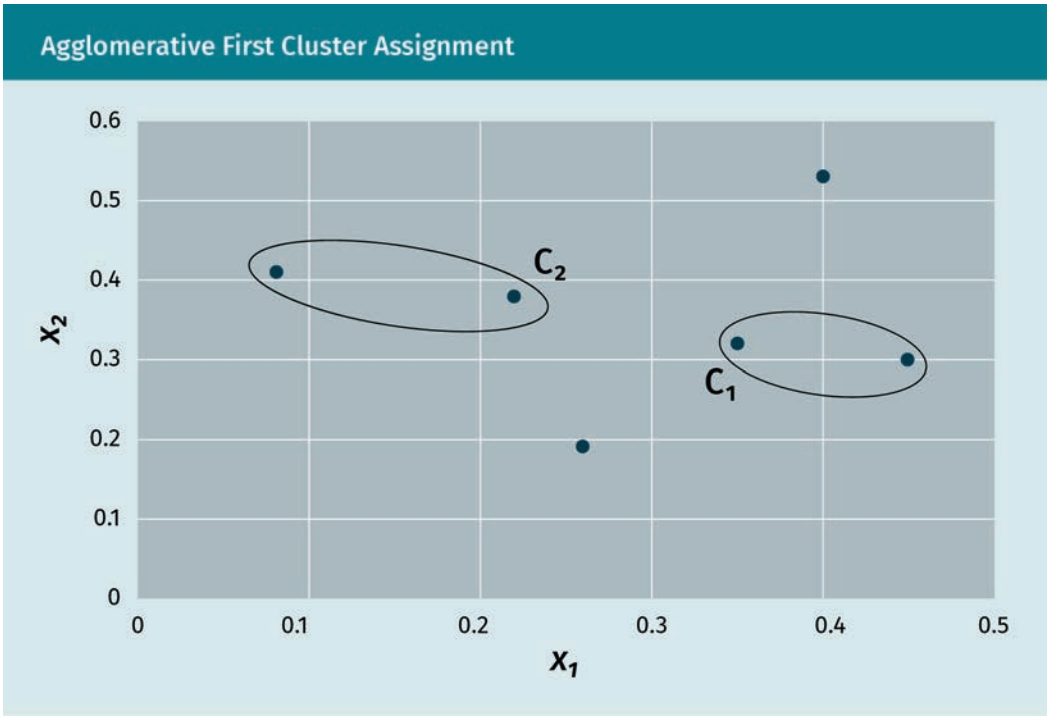
#	x_1	x_2
1	0.4	0.53
2	0.22	0.38
3	0.35	0.32
4	0.26	0.19
5	0.08	0.41
6	0.45	0.3



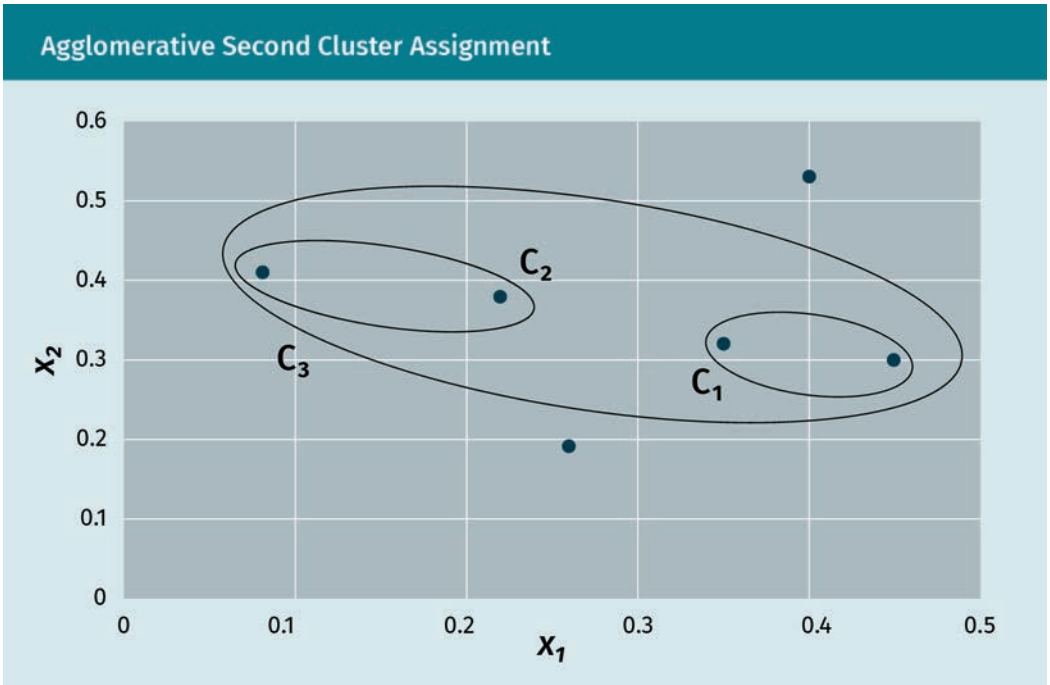
Each data record is assigned to a cluster, resulting in six clusters (leaves of the tree). The Euclidean distances between these data records form a symmetric matrix (also known as a proximity matrix).

Proximity Matrix						
#	1	2	3	4	5	6
1	0.00					
2	0.23	0.00				
3	0.22	0.15	0.00			
4	0.37	0.19	0.16	0.00		
5	0.34	0.14	0.28	0.28	0.00	
6	0.24	0.24	0.10	0.22	0.39	0.00

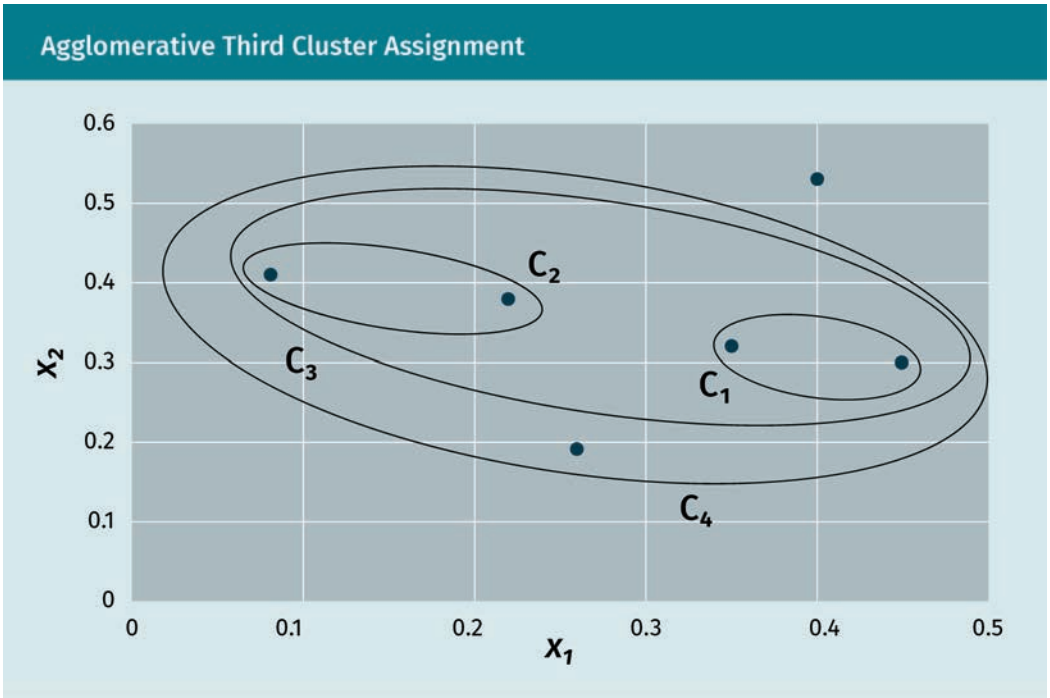
Here, the minimum distance is between data records #3 and #6, so they are merged into a single cluster C_1. The next minimum distance is between data records #2 and #5, so they are merged into another single cluster, C_2.



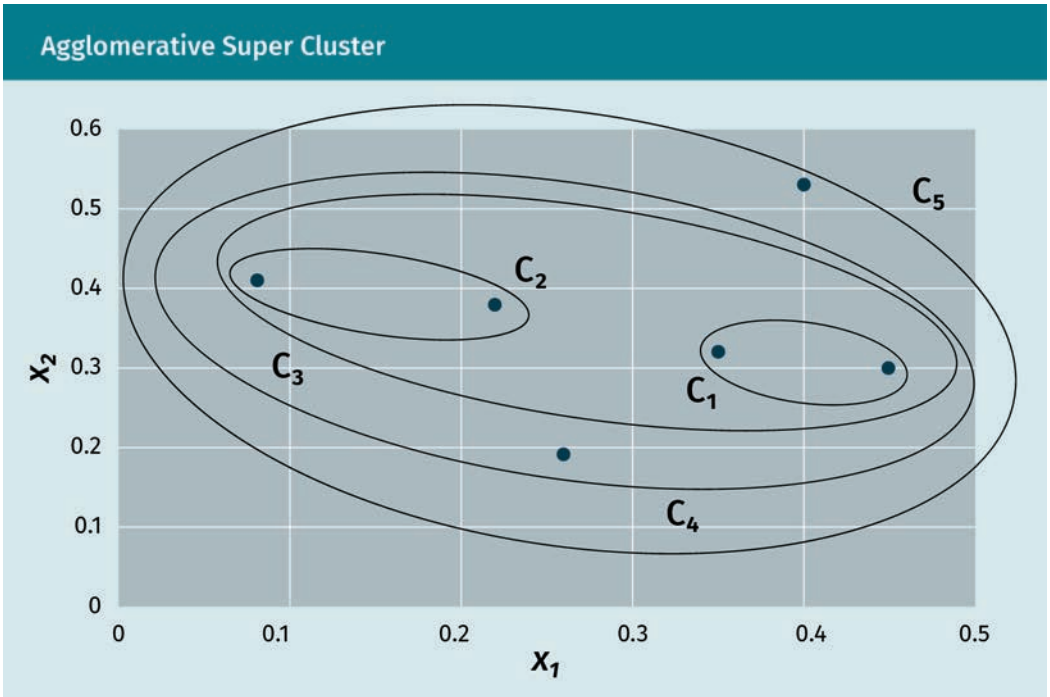
The third minimum distance is found between data records #2 (an element of C_2) and #3 (an element of C_1); therefore, these two clusters are merged into a single cluster, C_3.



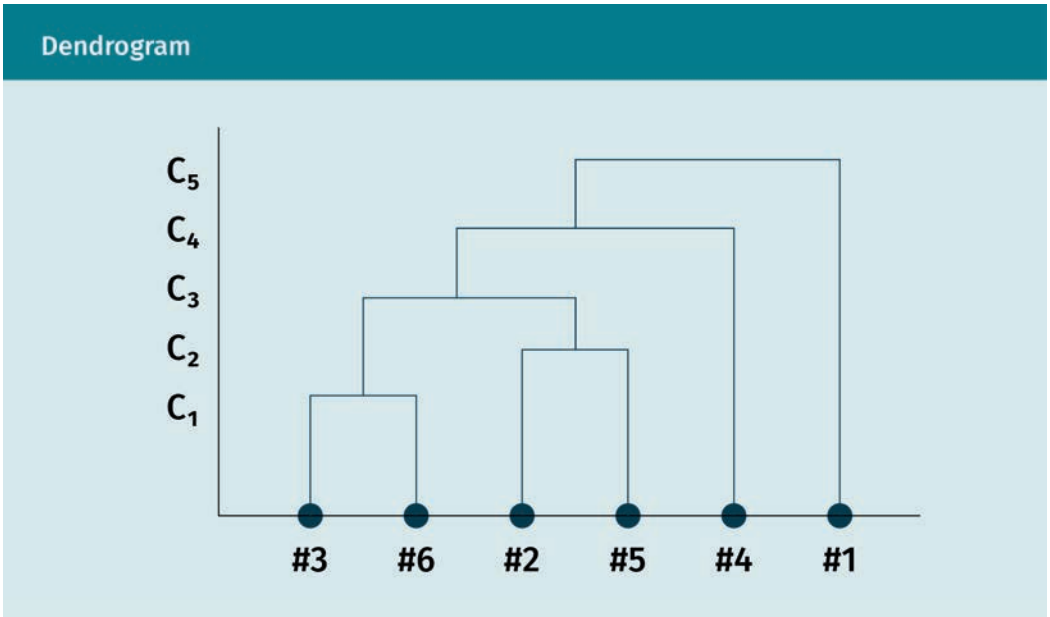
The fourth minimum distance is found between data records #4 and #3 (an element of C_3); therefore, single cluster C_4 is formed to include data record #4 and cluster C_3.



Finally, the super cluster C_5 is formed to include data record #1.



The dendrogram is shaped as:



A well-designed, standalone tool for clustering analysis can be accessed through the sklearn website.

5.3 Linear Regression

The objective of a regression model is to predict the value of a dependent (target) variable in a new situation, given the other independent (predictor) variables and the recorded target variable's behavior in previous situations. The regression model is a simplified representation of data performance given an underlying modeling assumption. In **linear regression**, the main assumption is that there are linear relationships between the data variables (Runkler, 2012). Building the model is an iterative process that results in a model that best relates the independent variables to the dependent ones. A model validation step is necessary to determine how well the model fits the data records.

Linear regression
This is a method for modeling linear relationships between a dependent variable and one or more independent variables.

Linear Regression Model

Let's assume we have a dataset with m variables $[x_1, x_2, ..., x_m]$ and n data records (1, 2, ..., n).

Example Dataset Table for Linear Regression

i	x_1	x_2	...	x_m	$\bar{y}$
1					

i	x_1	x_2	...	x_m	$\bar{y}$
2					
...					
n					

If a linear regression model has to be developed to predict the values of $\hat{y}$ (target variable) using the other variables [x_1, x_2, ..., x_m] (independent variables), this model is formed as:

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \cdots + w_mx_m$$

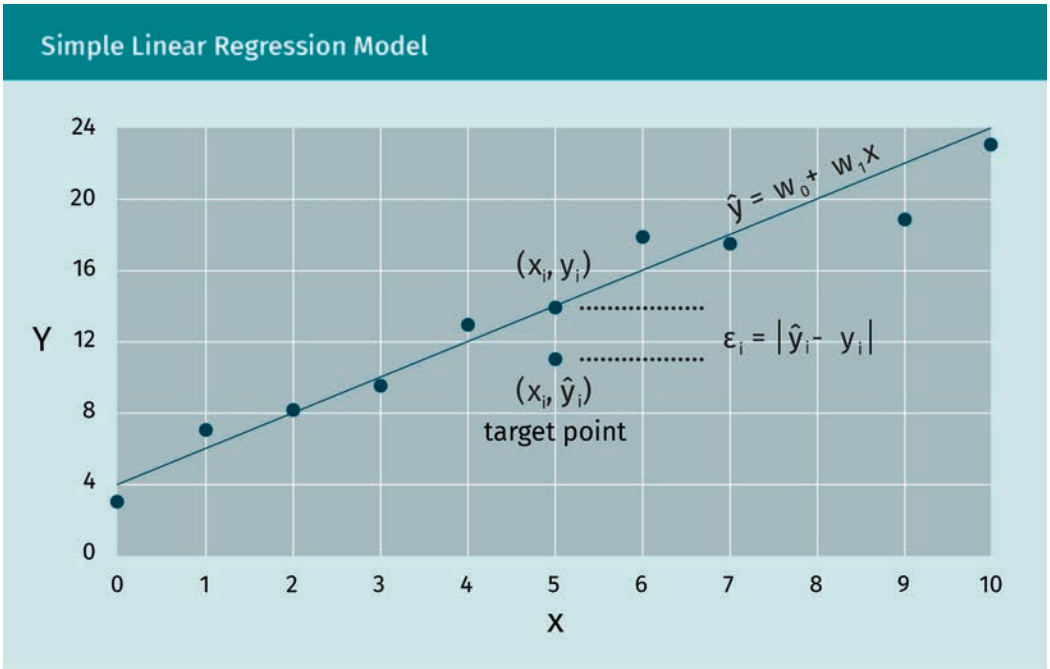
where w_0 is the bias and (w_1, w_2, ..., w_m) are the weights.

Unfortunately the output (y) differs slightly from ($\hat{y}$) by an error term (ε) because in the input dataset there is not an exact linear relationship between the dependent variables and the target variable.

Simple Linear Regression Model

For a simple case of one independent variable x, the linear regression model will be written as in the following equation and schematically shown as in the succeeding figure:

$$y = w_0 + w_1x$$



The bias w_0 represents the model intercept, and the weight w_1 represents the model slope. In such a scenario, the best model is the one which has the minimum error term values. Consequently, our task is to find the values of w_0 and w_1 which minimize the sum of the squared error $\left(\sum_{i=1}^{n} \varepsilon_i^2\right)$ (called the least-squares method). Mathematically, the error term has its minimum value at the instances where its derivative is zero (i.e., $\frac{\partial}{\partial w_0} \sum_{i=1}^{n} \varepsilon_i^2 = 0$ and $\frac{\partial}{\partial w_1} \sum_{i=1}^{n} \varepsilon_i^2 = 0$).

Therefore,

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (\hat{y} - y)^2 = \sum_{i=1}^{n} (\hat{y} - w_0 - w_1 x)^2$$

and

$$\frac{\partial}{\partial w_0} \sum_{i=1}^{n} \varepsilon_i^2 = 0$$

results in

$$2 \sum_{i=1}^{n} (\hat{y} - w_0 - w_1 x) \cdot (-1) = 0$$

$$\sum_{i=1}^{n} \hat{y} - w_0 \cdot n - w_1 \cdot \sum_{i=1}^{n} x = 0$$

$$w_0 = \frac{1}{n} \sum_{i=1}^{n} \hat{y} - \frac{w_1}{n} \sum_{i=1}^{n} x$$

and

$$\frac{\partial}{\partial w_1}\sum_{i=1}^{n}\varepsilon_i^2 = 0$$

results in

$$\begin{aligned}
&2\sum_{i=1}^{n}(\hat{y} - w_0 - w_1 x)\cdot(-x) = 0\\
&-2\sum_{i=1}^{n}\hat{y}x + 2w_0\sum_{i=1}^{n}x + 2w_1\sum_{i=1}^{n}x^2 = 0\\
&-2\sum_{i=1}^{n}\hat{y}x + 2\left(\frac{1}{n}\sum_{i=1}^{n}\hat{y} - \frac{w_1}{n}\sum_{i=1}^{n}x\right)\sum_{i=1}^{n}x + 2w_1\sum_{i=1}^{n}x^2 = 0\\
&-2\sum_{i=1}^{n}\hat{y}x + \frac{2}{n}\sum_{i=1}^{n}\hat{y}\sum_{i=1}^{n}x - \frac{2\cdot w_1}{n}\left(\sum_{i=1}^{n}x\right)^2 + 2w_1\sum_{i=1}^{n}x^2 = 0
\end{aligned}$$

$$w_1\left(\frac{-2}{n}\left(\sum_{i=1}^{n}x\right)^2 + 2\sum_{i=1}^{n}x^2\right) = 2\sum_{i=1}^{n}\hat{y}x - \frac{2}{n}\sum_{i=1}^{n}\hat{y}\sum_{i=1}^{n}x$$

$$w_1 = \frac{n\sum_{i=1}^{n}\hat{y}x - \sum_{i=1}^{n}\hat{y}\sum_{i=1}^{n}x}{-\left(\sum_{i=1}^{n}x\right)^2 + n\sum_{i=1}^{n}x^2}$$

The algorithm to construct the linear regression model which predicts a target variable (y) from the input data variables (x and $\overline{y}$) can be concluded in the following steps:

1. Calculate w_1:

$$w_1 = \frac{n\sum_{i=1}^{n}\hat{y}x - \sum_{i=1}^{n}\hat{y}\sum_{i=1}^{n}x}{-\left(\sum_{i=1}^{n}x\right)^2 + n\sum_{i=1}^{n}x^2}$$

2. Calculate w_0:

$$w_0 = \frac{1}{n}\sum_{i=1}^{n}\hat{y} - \frac{w_1}{n}\sum_{i=1}^{n}x$$

3. Insert w_1 and w_0 into the linear model equation $y = w_0 + w_1 x$
4. To use the developed model for forecasting the value of the target variable at a new data entry $\hat{x}$, substitute in the model equation $\hat{y} = w_0 + w_1 \hat{x}$

Simple Linear Regression Example

Develop a linear regression model to predict the revenue of a company in 2017, given its revenues from 2011 to 2015.

Company Revenues	
x (year)	$\bar{y}$ (million USD)
2011	50
2012	54
2013	58
2014	55
2015	60

First we calculate $\sum_{i=1}^{n} x$, $\sum_{i=1}^{n} x^2$, $\sum_{i=1}^{n} \hat{y}x$, and $\sum_{i=1}^{n} \hat{y}$:

Calculations			
x (year)	$\hat{y}$ (million USD)	x^2	$x\hat{y}$
2011	50	4044121	100550
2012	54	4048144	108648
2013	58	4052169	116754
2014	55	4056196	110770
2015	60	4060225	120900

x (year)	$\hat{y}$ (million USD)	x^2	$x\hat{y}$
$\sum_{i=1}^{n} x = 10065$	$\sum_{i=1}^{n} \hat{y} = 277$	$\sum_{i=1}^{n} x^2 = 20260855$	$\sum_{i=1}^{n} \hat{y}x = 557622$

1. $w_1 =$

$$\frac{n\sum_{i=1}^{n} \hat{y}x - \sum_{i=1}^{n} \hat{y} \sum_{i=1}^{n} x}{-\left(\sum_{i=1}^{n} x\right)^2 + n\sum_{i=1}^{n} x^2} = \frac{5 \cdot 557622 - 277 \cdot 10065}{-(10065)^2 + 5 \cdot 20260855} = \frac{105}{50} = 2.1$$

2. Calculate w_0:

$$w_0 = \frac{1}{n}\sum_{i=1}^{n} \hat{y} - \frac{w_1}{n}\sum_{i=1}^{n} x = \frac{277}{5} - \frac{2.1 \cdot 10065}{5} = -4171.9$$

3. Insert w_1 and w_0 into the linear model equation:

$$y = w_0 + w_1 x = -4171.9 + 2.1 \cdot x$$

4. At $x = 2017$, the revenue is predicted to be $y = -4171.9 + 2.1 \cdot 2017 = 63.8$ million USD.

Multiple Linear Regression Model

For a dataset with more than one independent variable (x_1, x_2, ... x_m), the model will be extended to include one term for each variable, as given in the general linear regression equation:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$$

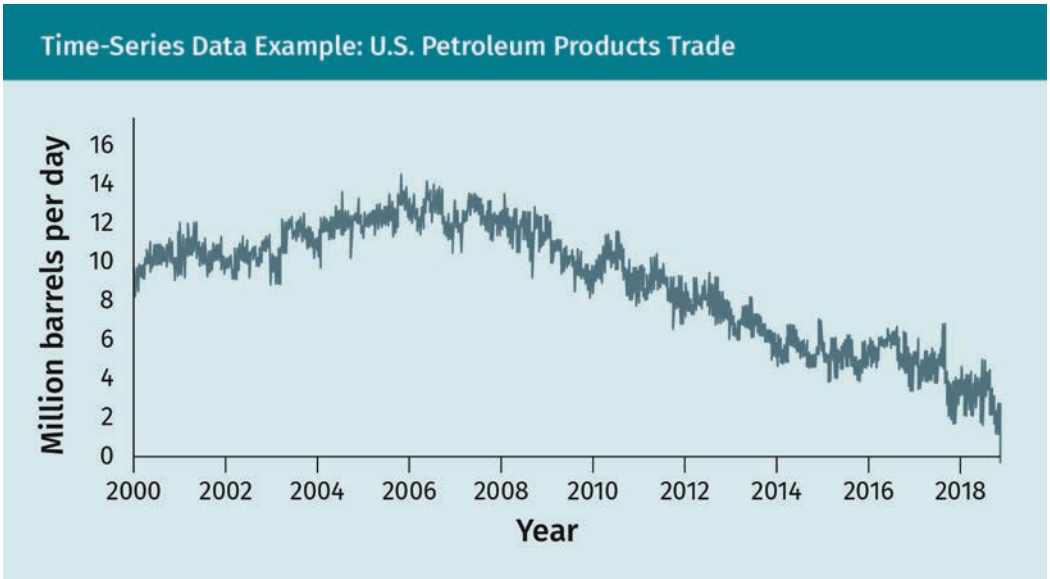
The derivative of the error term with respect to each weight is set to zero (i.e., $\frac{\partial}{\partial w_0}\sum_{i=1}^{n} \varepsilon_i^2 = 0$, ..., $\frac{\partial}{\partial w_m}\sum_{i=1}^{n} \varepsilon_i^2 = 0$), resulting in the calculation of the weights (w_0, w_1, w_2, ..., w_m). The values of the weights provide a descriptive overview of the correlation between the target variable and each independent variable. For example, a large value for w_m implies that the target y is highly correlated to the variable x_m and vice versa.

However, for a dataset with many independent variables, the assumption of a linear relationship between the target variable and other variables becomes weak. Consequently, nonlinear regression models will produce more accurate predictions.

5.4 Time-Series Forecasting

The development of forecasting models for time-series data is the focus of many business organizations. **Forecasting models** offer businesses a chance to estimate future performance and prepare for any issues the model raises. Examples of time-series data that require predictive analysis and the application of forecasting models are stock price, sales, traffic flow, and petroleum production.

Forecasting model
This is a model for forecasting (predicting) future events based on data and information gleaned from the past.



A simple analysis of time-series data is noting if the observations tend to increase (or decrease) over time; if there are regularly repeating patterns over time; and if there are outliers in the observations.

Forecasting uses the observed input and output instances ($\{t_1, t_2, ..., t_n\}$, $\{y_1, y_2, ..., y_n\}$) to predict the expected output sequence $\{y_{n+1}, y_{n+2}, ...\}$ without knowing the expected input instances $\{t_{n+1}, t_{n+2}, ...\}$. One important difference between time-series forecasting and basic regression analysis is that individual data records depend (to some degree) on the previous data records. This means that each record is dependent on its history, and a forecasting technique needs to take this into consideration. Therefore, it is necessary to order observations with respect to time instances.

A popular linear forecasting technique is the autoregressive method (AR) which assumes expected output is a linear function of some past outputs. However, if the underlying relationship is nonlinear, the AR approach yields suboptimal results. Hence, to obtain better results for nonlinear data, the AR method is upgraded to include moving average (ARMA) and integral terms (ARIMA).

Concept of Stationary

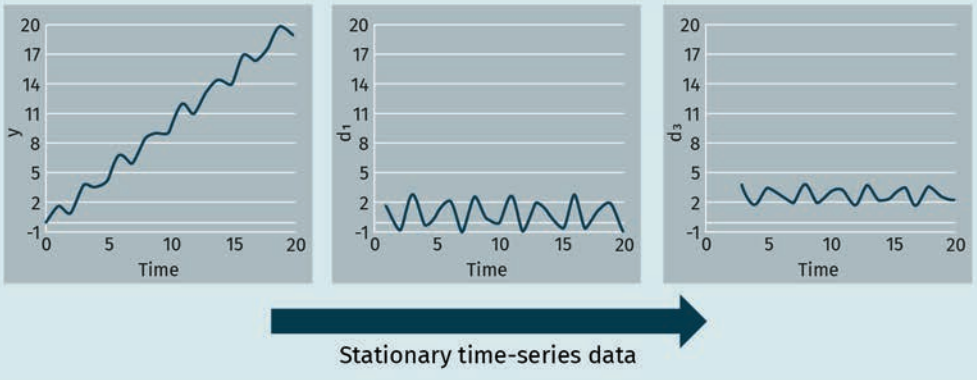
Stationary Time-Series

Data in a stationary time-series has a constant mean and standard deviation over time.

In order to apply a forecasting model to time-series data, the data should be **stationary** over time, because only with stationary data can the model correctly self-predict its future response from past data points. While most time-series data are not considered stationary, it is easy to convert them into stationary data using the differencing concept ($d_{\Delta t}$), where $d_{\Delta t}$ is the difference between every two data points having Δt interval.

Concept of Stationary

t	y	d_1 $(y_t - y_{t-1})$	d_2 $(y_t - y_{t-2})$	d_3 $(y_t - y_{t-3})$
0	0.000			
1	1.650	1.650		
2	1.012	-0.638	1.012	
3	3.851	2.839	2.201	3.851
4	3.695	-0.156	2.683	2.045
5	4.612	0.917	0.761	3.600
6	6.894	2.282	3.199	3.043
7	6.029	-0.865	1.417	2.334
8	8.581	2.551	1.687	3.968
9	9.088	0.508	3.059	2.194
10	9.285	0.197	0.705	3.256
11	11.998	2.713	2.909	3.417
12	11.199	-0.799	1.914	2.110
13	13.219	2.021	1.222	3.934
14	14.468	1.248	3.269	2.470
15	14.070	-0.398	0.850	2.871
16	16.945	2.876	2.478	3.726
17	16.494	-0.452	2.424	2.026
18	17.824	1.330	0.879	3.754
19	19.774	1.950	3.280	2.828
20	19.000	-0.774	1.176	2.507



It is worth mentioning that there is an optimum value for Δt, which results in a completely stationary form of the time-series data.

Autoregressive (AR) Model

The autoregressive model is a linear model developed to predict the value of an observation at the very next point in time using linear combinations of its values at previous time instances. It is equivalent to a simple linear regression model, and because it uses data from the same variable at past points in time, it is called autoregressive (i.e., self-regression).

AR(n) stands for an autoregressive model of order n, which indicates that n previous observations (i.e., **lag(n)**) are used in the prediction of the next observation:

Lag(n)
This is the backshift of a time-series by n time steps.

$$y_t = p_0 + p_1 y_{t-1} + p_2 y_{t-2} + \cdots + p_n y_{t-n} \quad + \quad \varepsilon_t$$

where $\{p_0, p_1, ..., p_n\}$ are the model coefficients and ε_t is a white noise term $WN(0, \sigma^2)$.

Moving Average (MA) Model

The moving average model predicts future observations:

$$y_t = q_0 + q_1 \varepsilon_{t-1} + q_2 \varepsilon_{t-2} + \cdots + q_n \varepsilon_{t-n}$$

where ε_{t-n} are the white noise error terms $WN(0, \sigma^2)$. The elements $\{q_0, q_1, ..., q_n\}$ are the model coefficients.

Autocorrelation

The correlation coefficient between two variables refers to the degree to which their relationship is linear. The coefficient is a number is between –1 (indicating a negative correlation) and 1 (indicating a positive correlation). If both variables increase (or decrease) together, there is positive correlation; if the variables move in opposite directions, there is negative correlation (Brownlee, 2017).

We can calculate the correlation between the forecasted variable and its values at previous lags. The stronger the correlation between the forecasted variable and its value at a specific lag, the more weight the autoregressive model can put on this specific lagged value. Because the correlation is performed between the variable and itself at previous lags, it is called an autocorrelation. Hence, the correlation coefficient between a variable and its different lags' values is calculated to recognize which lags are good candidates to be used in the developed autoregressive model.

The autocorrelation coefficient ACF(n) at lag(n) is given by the following equation:

$$\mathrm{ACF(n)} = \frac{\mathrm{C(y_t,\, y_{t-n})}}{\sqrt{\mathrm{V(y_t)} \cdot \mathrm{V(y_{t-n})}}}$$

where $\mathrm{C(y_t,\, y_{t-n})}$ is the covariance coefficient between $\mathrm{y_t}$ and $\mathrm{y_{t-n}}$ and $\mathrm{V(y_t)}$ ist the cariance of $\mathrm{y_t}$, and $\mathrm{V(y_t)} = \mathrm{C(y_t,\, y_t)}$.

Partial Autocorrelation

The partial autocorrelation function ($\mathrm{PACF_k}$) at lag(k) is the autocorrelation between $\mathrm{y_t}$ and $\mathrm{y_{t-k}}$ that is not accounted for by the autocorrelations from the 1st to the $\mathrm{(k-1)^{st}}$ lags. It is obtained by solving the following equation:

$$\begin{pmatrix} \mathrm{ACF(0)} & \mathrm{ACF(1)} & \cdots & \mathrm{ACF(k-1)} \\ \mathrm{ACF(1)} & \mathrm{ACF(0)} & \cdots & \mathrm{ACF(k-2)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{ACF(k-1)} & \mathrm{ACF(k-2)} & \cdots & \mathrm{ACF(0)} \end{pmatrix} \begin{pmatrix} \mathrm{PACF_1} \\ \mathrm{PACF_2} \\ \vdots \\ \mathrm{PACF_k} \end{pmatrix} = \begin{pmatrix} \mathrm{ACF(1)} \\ \mathrm{ACF(2)} \\ \vdots \\ \mathrm{ACF(k)} \end{pmatrix}$$

Autoregressive Integrated Moving Average (ARIMA) Model

ARIMA models mix both the (AR) and (MA) models with integrated parameters in one model to obtain a better understanding of time-series data and/or to forecast future data points in the series (Dalinina, 2017). An integrated parameter is the degree of differencing which is performed on the dataset to transform it into a stationary time-series. The notation of ARIMA is ARIMA(p, d, q), where p stands for the number of (AR) terms, d is the degree of differencing, and q is the number of (MA) terms. The general form of the model is:

$$\begin{aligned} \mathrm{y_t} &= \mathrm{costant + weighted\ sum\ of\ the\ previous\ p\ values\ of\ y} \\ &\quad + \mathrm{weighted\ sum\ of\ the\ previous\ q\ forecast\ errors} \\ \mathrm{y_t} &= \mathrm{c + p_1 y_{t-1} + \cdots + p_n y_{t-n} + q_1 \varepsilon_{t-1} + \cdots + q_m \varepsilon_{t-n}} \end{aligned}$$

The questions now are: How many terms to use for (p)? How many terms for (q)? The simple answer is to check some combinations of (p) and (q) that often come up in practice. But the systematic procedure to estimate them is to look at the plots of autocorrelation and partial autocorrelation functions of the time-series and consider the following rules (Nau, 2014):

1. "If the ACF plot cuts off sharply at lag(k) (i.e., if the autocorrelation is significantly different from zero at lag(k) and extremely low in significance at the next higher lag and the ones that follow), while there is a more gradual decay in the PACF plot (i.e., if the decay is significance beyond lag(k) is more gradual), then set $q = k$ and $p = 0$."
2. "If the PACF plot cuts off sharply at lag(k) while there is a more gradual decay in the ACF plot beyond lag(k), then set $p = k$ and $q = 0$."
3. "If there is a single spike at lag(1) in both the ACF and PACF plots, then set $p = 1$ and $q = 0$ if the spike is positive, and set $p = 0$ and $q = 1$ if it is negative."

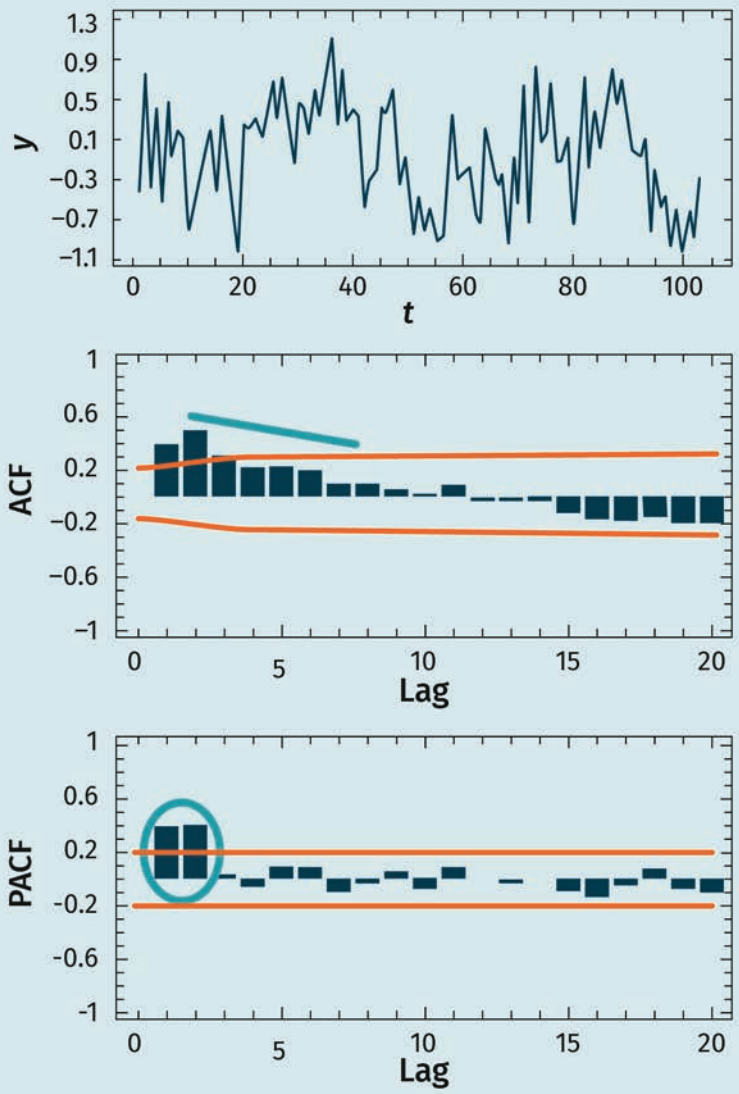
To determine that a value is a "spike," it should exceed the 95 percent confidence interval.

In the following figures we will present three examples for different time-series datasets and their ACF and PACF plots to show how these plots could estimate the terms of the designed ARIMA models.

In the first example, the top plot is the input time-series data, which looks stationary over time; therefore, there is no need for a differencing process. The bottom plot is for the PACF values, which cut off sharply at lag(2), and the middle plot is for the ACF values, which gradually decay after lag(2). Therefore, we can set $p = 2$ and $q = 0$, resulting in the ARIMA(2, 0, 0) model here:

$$y_t = c + p_1 y_{t-1} + p_2 y_{t-2}$$

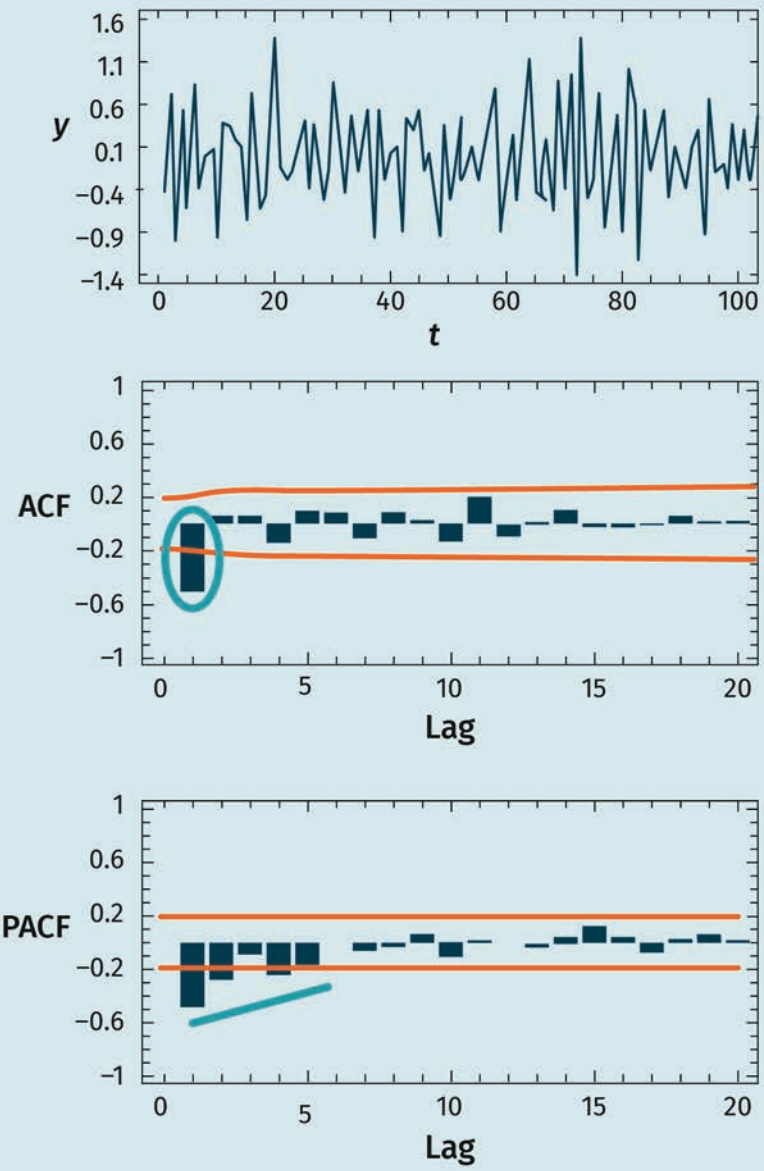
Example (1): ARIMA (2, 0, 0) Model



In example (2), the top plot is an input stationary time-series data. The middle plot is for the ACF values, which cut off sharply at lag(1), while the bottom plot is for the PACF values, which gradually decay after lag(1). Therefore, we can set $\mathrm{p} = 0$ and $\mathrm{q} = 1$, resulting in the ARIMA(0, 0, 1) model here:

$$\mathrm{y_t} = \mathrm{c} + \mathrm{q_1}\varepsilon_{\mathrm{t}-1}$$

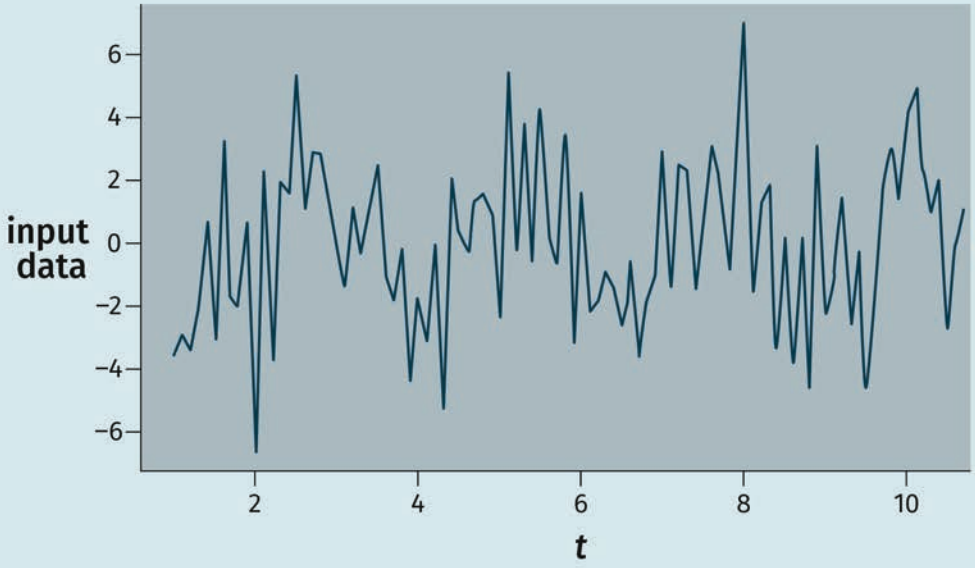
Example (2): ARIMA (0, 0, 1) Model

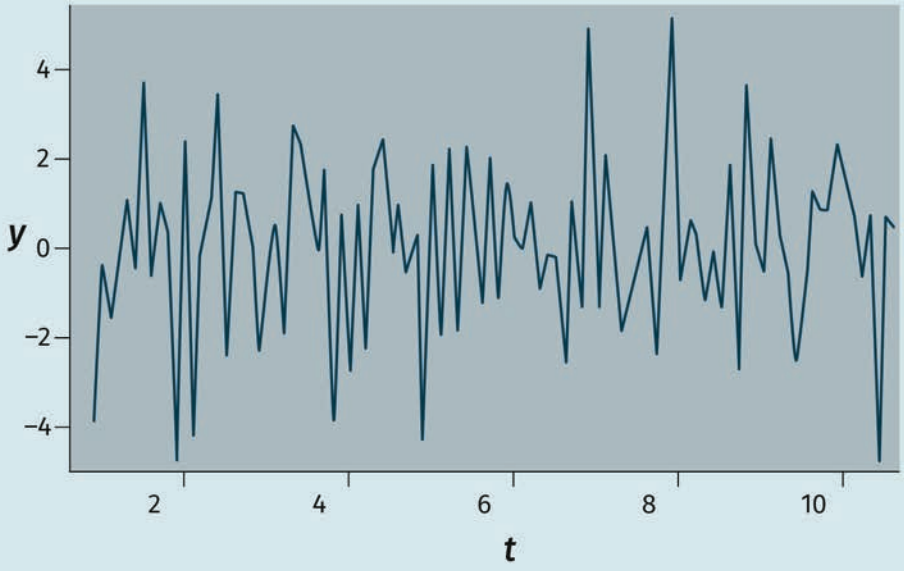


In example (3), the top plot is an input nonstationary time-series data. A one degree differencing is performed, resulting in almost stationary data (i.e., $d = 1$). The ACF plot of this differencing data shows significant spikes up to lag(2), and the PACF plot shows spikes up to lag(5). Therefore, an ARIMA(5, 1, 2) model is estimated, as given here:

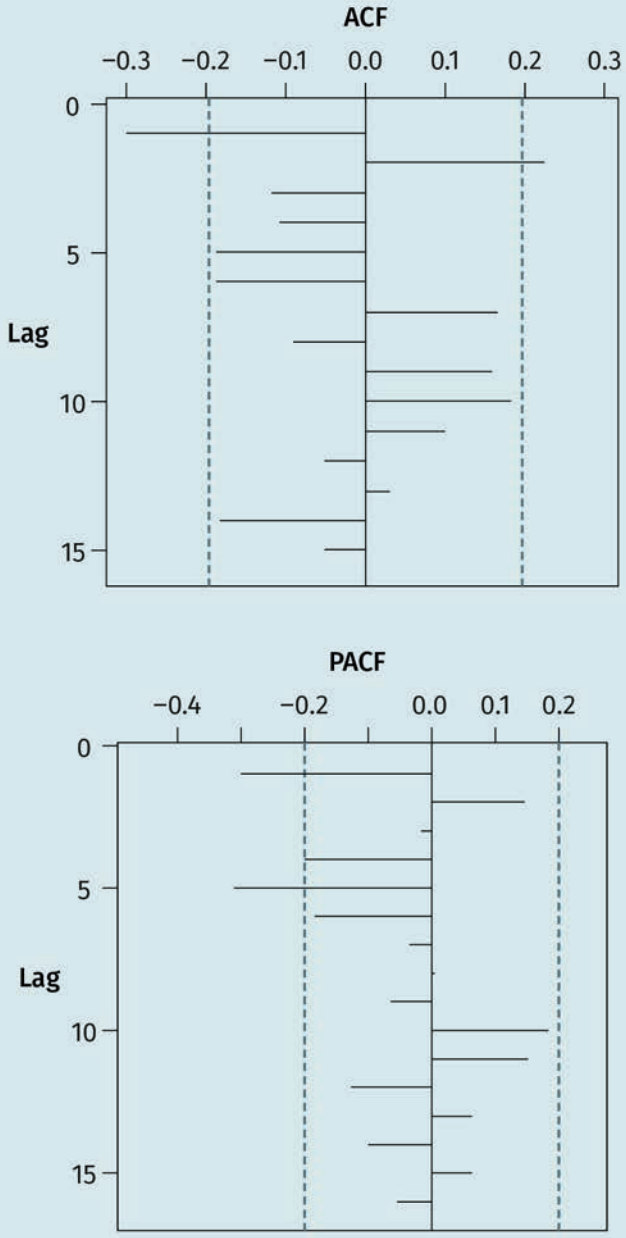
$$y_t = c + p_1 y_{t-1} + p_2 y_{t-2} + q_1 \varepsilon_{t-1} + q_2 \varepsilon_{t-2} + q_3 \varepsilon_{t-3} + q_4 \varepsilon_{t-4} + q_5 \varepsilon_{t-5}$$

Example (3): ARIMA (5, 1, 2) Model





Example (3): ARIMA (5, 1, 2) Model (ACF/PACF)



Once a good estimation of the number of terms in the developed model y_t is made, its unknown coefficients on the right-hand side are obtained using the input time-series data. Afterwards, we get the residuals time-series (i.e., differences between the input time-series and the model's forecasted time-series), R_t. If the ARIMA model was properly developed with the correct number of terms, there should be no significant spikes in the ACF and PACF plots of R_t. If there are significant spikes at higher order lags (e.g.,

the model is ARIMA(2, 0, 0) and there is a significant spike at lag(5) in the residual analysis), it can be the case that the developed model is correct, although a model with one extra differencing process can still be tried.

Meanwhile, if there is a significant spike at a lower order lag during the residual analysis (e.g., at lag(2) while the model is ARIMA(1, 0, 0) or ARIMA(0, 0, 1)), we apply the following rules:

1. If the spike is in the ACF plot, then we increase q by 1 and refit the model.
2. If the spike is in the PACF plot, then we increase p by 1 and refit the model.

In practice, the values for q or p are not larger than 3 in any developed ARIMA model for a business application. It is advised to avoid using "mixed" models in which there are both q and p terms. Moreover, if a decision is made to add additional terms to the developed model on the basis of the residual analysis recommendation, we change one parameter at a time (i.e., increase q by 1 or increase p by 1, then refit the model to see what the effect is before going any further). It is important to note that time-series data may first need nonlinear transformation, such as logging and/or raising to some power, in order to be converted into a form with consistent distribution over time and symmetry in appearance.

Seasonal Autoregressive Integrated Model (SARIMA)

Although ARIMA is one of the most widely-used forecasting methods for time-series data, it cannot handle datasets with seasonal components. Seasonal time-series data are cyclical, repeating after a specific period of time. For example, monthly weather data are repeated every 12 months. Therefore, the seasonal autoregressive integrated model (SARIMA) is required.

The formula for the SARIMA model is given as SARIMA$(p, d, q)(P, D, Q)s$, where (p, d, q) are the three parameters of ARIMA(p, d, q), respectively. (P, D, Q) are similar to the nonseasonal components of the model, but they involve backshifts of the seasonal period. The number of time steps for a single seasonal period is denoted by s. There are many online tools for developing SARIMA models. For example, Python's statsmodels library supports the complete designing, fitting, and forecasting of a SARIMA model for any input seasonal time-series dataset. These online tools can also be used for SARIMAX models, which permit the existence of an exogenous variable (X) in the dataset. The (X) variable may be an external variable which influences the time-series variations and needs to be considered during the analysis.

5.5 Transformation Approaches

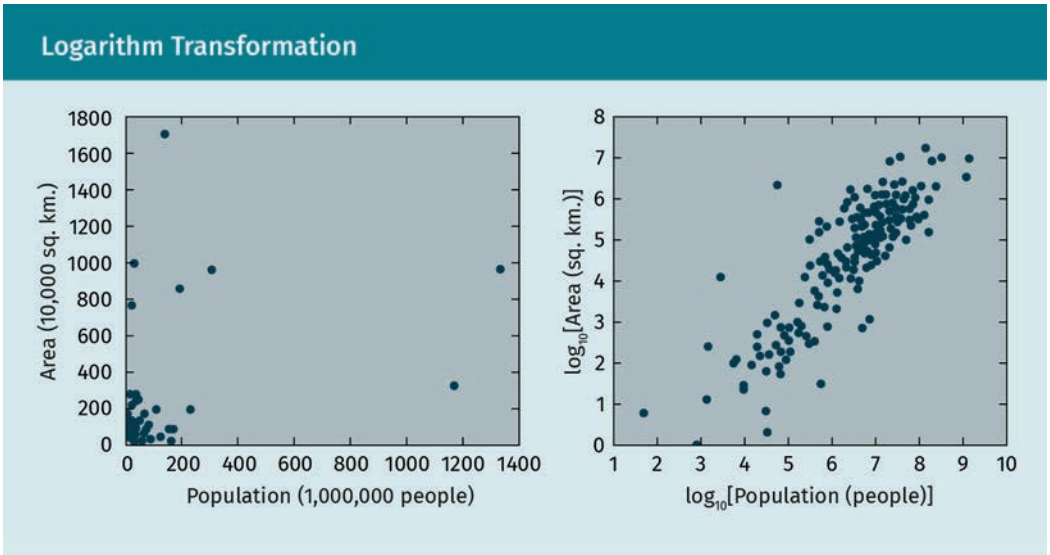
The dataset transformation is the application of a mathematical function (f) to each data variable (x_i); as such, (x_i) is replaced with a renovated variable (X_i).

$$X_i = f(x_i)$$

The objective of transforming a dataset is to improve its interpretability and transform its variables to a new space (e.g., moving from Cartesian coordinates to radial coordinates) where more relevant variables can be easily extracted and utilized.

Logarithm Transformation

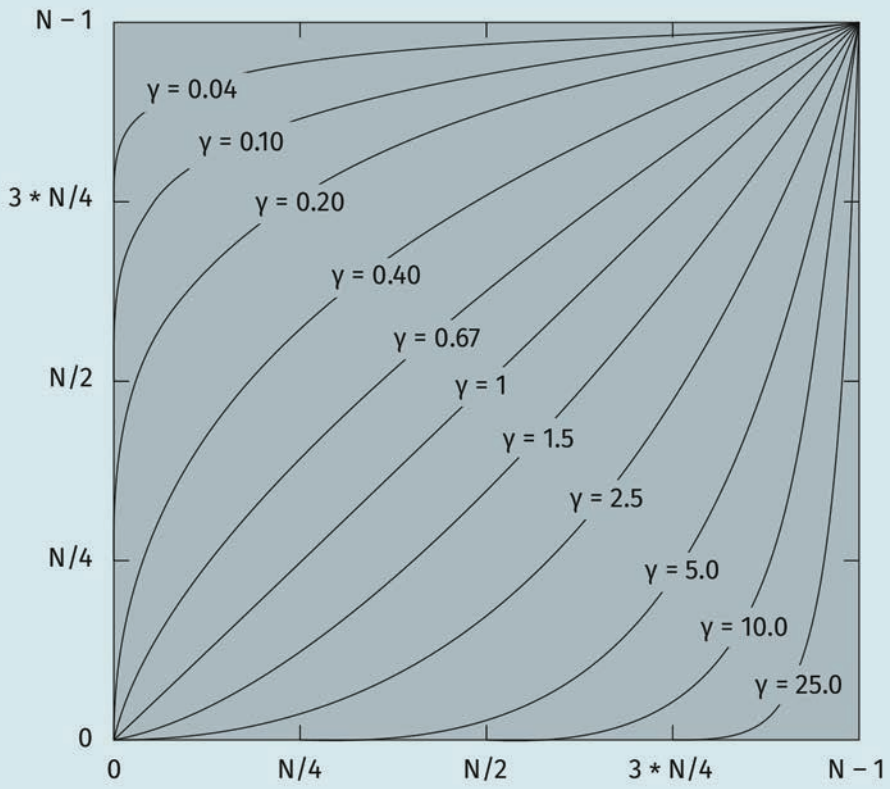
The logarithm transformation, $= \log(x)$, is usually applied to linear regression problems when the developed model fails to hold the linearity between the input variables. For example, if the input dataset is for a country's population, it is common to transform the variables using a logarithm function to obtain an approximately linear data variation.



Power Law Transformation

The power law transformation, $X = x^{\gamma}$, represents a family of transformations based on a nonnegative parameter (γ). The value of γ is initially estimated and then continuously updated during the training phase of the model-building process until the highest performance accuracy is achieved.

The Power Law Transformation



Reciprocal Transformation

The reciprocal transformation, $X = \frac{1}{x}$, changes the shape of a variable to its inverse. This can be particularly useful when the reverse of a variable makes more sense for the data analysis. For example, data are given as the number of students per teacher, but it is more relevant to have the number of teachers per student.

Radial Transformation

The radial transformation focuses on the distance between the value of a variable and the origin. It combines two variables (x_1) and (x_2) and transforms them into the radial coordinates of radius ($r = \sqrt{x_1^2 + x_2^2}$) and angle ($\theta = \tan^{-1}\frac{x_1}{x_2}$).

Discrete Fourier Transform

The Fourier transform is applied to transfer a variable of a dataset from its traditional domain (e.g., variable (x) versus time (t)) to its frequency domain (i.e., spectrum of variable (x) versus frequency (f)). Therefore, the Fourier transform determines which frequencies can represent the distribution of a given variable. The Fourier transform follows the formula:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi i n k / N}$$

where N is the length of the selected frequency band.

Summary

Principal component analysis (PCA) is typically applied to a dataset in order to rank the data variables according to their relevancy to the underlying data values. This ranking is achieved by searching for the direction of the data's maximum variance and constructing the first principal component. Then the second maximum variance is determined, which is where the second principal component is placed. The process is repeated as many times as the number of data variables. The first few principal components may be used in place of the original data variables, and by doing this, the dataset is reduced in dimensionality.

In a cluster analysis, data records are grouped into clusters according to their level of similarity. The two routinely-employed clustering approaches are K-means and agglomerative clustering.

The prediction of a new dataset of values is determined by general regression analysis or—if time is an independent variable of the underlying dataset—the time-series analysis. The developed prediction model is an approximation for the relation between the target variable and the independent variables. For linear relations, the simple linear regression, multiple linear regression, and ARIMA models are developed to achieve the prediction task.

Before designing a prediction model, the dataset variables may need to be transformed and scaled to show their impact on the data variations. Common transformation techniques are logarithm, power law, reciprocal, radial, and Fourier.

Knowledge Check

Did you understand this unit?

Now you have the chance to test what you have learned on our Learning Platform.

Good luck!

Unit 6

Selected Artificial Intelligence Techniques

STUDY GOALS

On completion of this unit, you will have learned ...

... data classification by support vector machines.

... the feedforward neural network structure.

... the back propagation algorithm in neural networks.

... how to develop an artificial neural networks prediction model.

... recurrent networks and reinforcement learning.

... basics about genetic algorithms, fuzzy logic, and Naïve Bayes classification.

6. Selected Artificial Intelligence Techniques

Introduction

Beside regression, the second approach in supervised learning is classification where a prediction model is developed based on the input dataset in order to estimate the class of a new data record. In this unit, we will discuss a common classification method: support vector machines (SVM). We will also provide a detailed discussion on the well-known nonlinear regression model, the artificial neural network (ANN).

A short overview will be given on fuzzy logic and genetic algorithms as well as Naïve Bayes classification. Fuzzy logic is helpful in obtaining a better and broader idea of a dataset and determining if it has missing values. Genetic algorithms are applied to search for and optimize possible solutions to the dataset problems in an adaptive manner (i.e., reduce uncertainty over time by process monitoring). Meanwhile, Naïve Bayes classification is another common approach based on the Bayes theorem to classify data records according to conditional probability criteria.

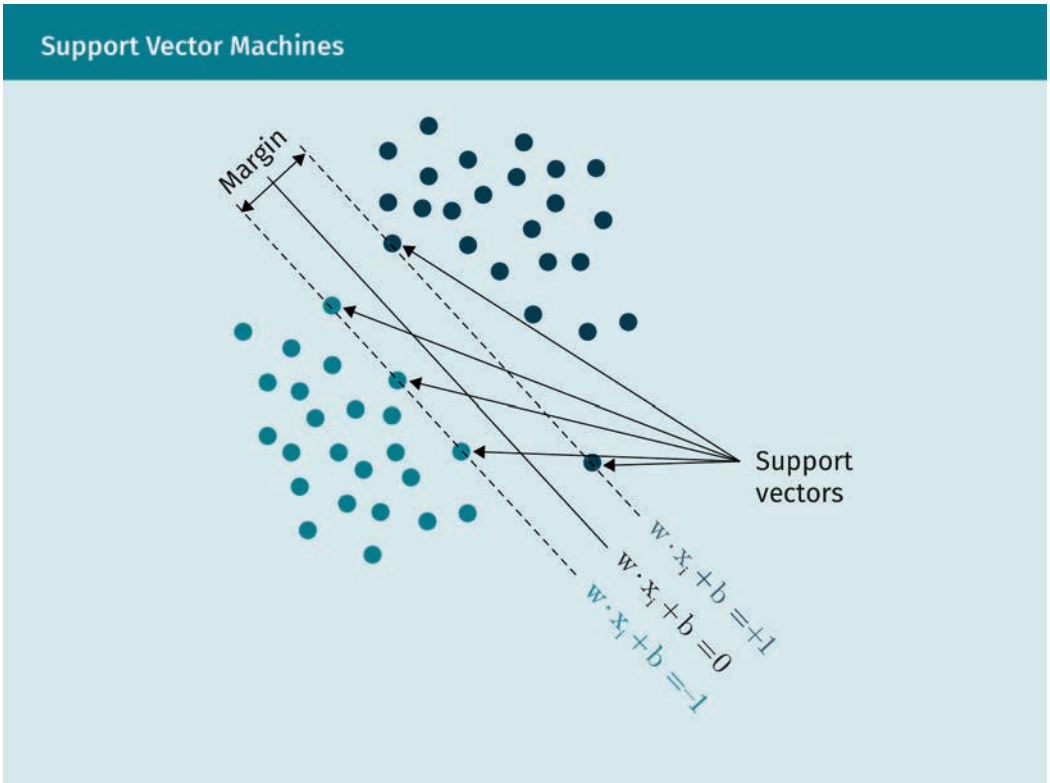
6.1 Support Vector Machines

Support vector machines
This is a supervised learning algorithm in machine learning that is typically used in classification problems.

The **support vector machines** (SVM) technique can be used for both classification and regression, but the discussion here will be limited to the implementation of SVM in classification problems. SVM is a binary linear classification technique where the classification rule is to develop a linear function of the input dataset variables $\{x_{1k}, x_{2k}, ..., x_{Mk}\}$, with $k = 1, 2, ..., N$, and where N and M are the number of data records and data variables, respectively (Polson & Scott, 2011). This classification function can be simply formulated as the linear equation:

$$w \cdot x_i + b = 0$$

If the dataset has two classes (e.g., rich people/poor people, tall students/short students), the classification line will separate the dataset according to class, with one class on either side of the line.



The separating channel (i.e., the channel which has the classification line on its center line and is bounded by support vectors on both sides) is generated by the classification line and defined by the two lines parallel to it and equidistant from both sides, namely:

$$w \cdot x_i + b = +1$$

and

$$w \cdot x_i + b = -1$$

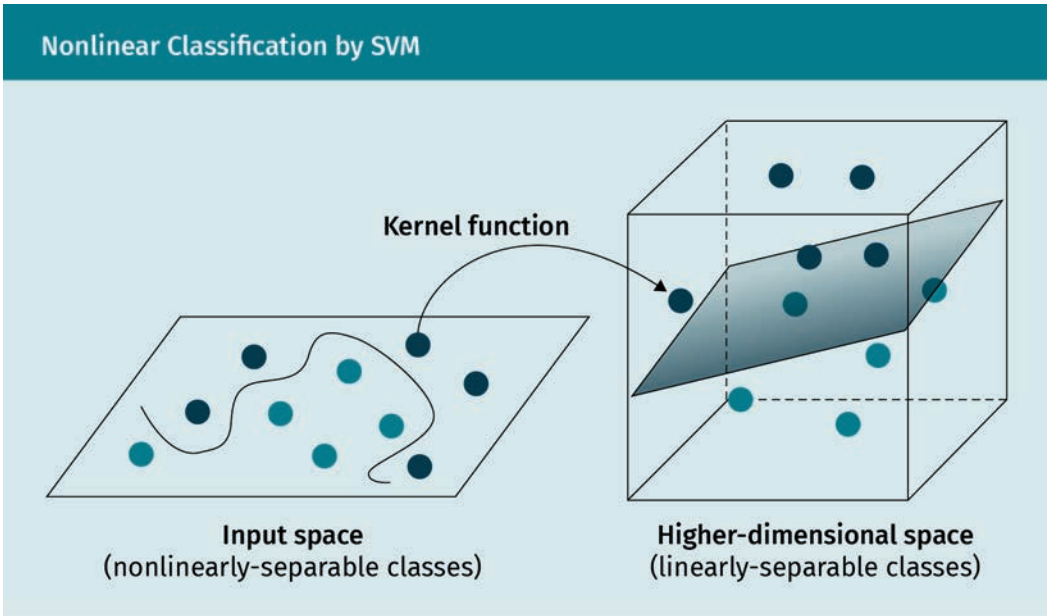
The support vectors are the data records x_i that lie on either side of the separating channel. The margin (l) is the distance between the two sides, which can be calculated as the distance between the support vectors on the right side and the support vectors on the left side.

$$l = x_i^{+1} - x_i^{-1}$$
$$\therefore l = \frac{+1 - b}{w} - \frac{-1 - b}{w} = \frac{2}{||w||}$$

The SVM technique seeks the maximum margin (l) to obtain the optimum separation between the two classes. Therefore, the problem is transformed into an optimization problem which solves the classification equation and yields maximum $\left(\frac{2}{||w||}\right)$.

Kernel Trick

The strategy of SVM in dealing with nonlinearly separable datasets is to use the Kernel trick (Wenzel et al., 2017). The idea is to project all data points onto a higher dimensional space in which the dataset is linearly separable. Instead of actually making projections to higher dimensions, SVM only considers the relative distance between the data points in the "virtual" projection space, as given by an appropriate Kernel function. Common Kernel functions include polynomial, sigmoid, and radial.



There are many online software packages that will perform SVM classification on any dataset. The library for support vector machines, LIBSVM, is an excellent online resource and includes helpful software downloads (Chang & Lin, 2011). The "Scikit-Learn" tool in Python is also a great online resource that is appropriate for clustering and regression.

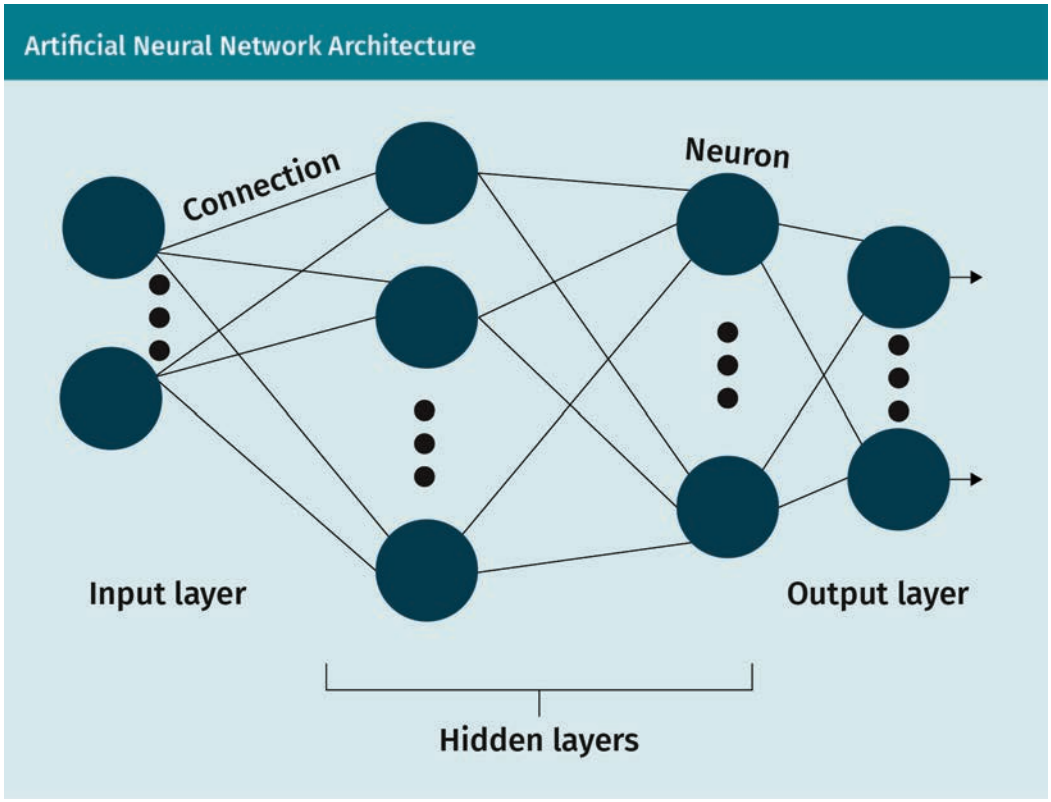
6.2 Artificial Neural Networks

The human brain includes billions of interconnected neurons that transmit signals to one another. The sending and receiving of signals between two neurons depends on the strength of the connection between them. The neurons and their interconnections work within the full neural network system to learn concepts, recognize objects, and make new predictions. Memories are stored within the connections, and a memory pattern is a rough characterization of how and to what degree neurons are connected.

The purpose of developing an **artificial neural network** (ANN) is to produce an artificial system capable of performing sophisticated calculations similar to the human brain. The response is encoded in how and to what degree various neurons are connected.

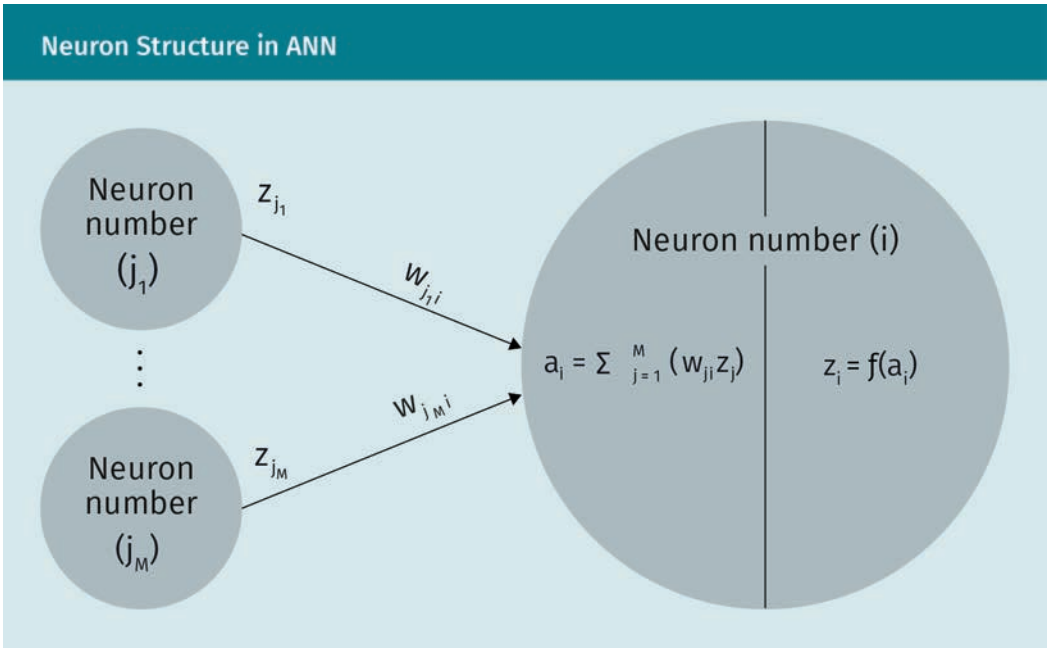
The basic network architecture comprises many layers of neurons: the input layer is for the input values of the dataset variables, and the output layer produces the value of the target variable. The intermediate layers are referred to as the hidden layers. The application of artificial neural networks to learning tasks with cascading hidden layers is called deep learning.

Artificial neural network
This is a computing network based on the neural networks of animal brains. It contains nodes and arrows.



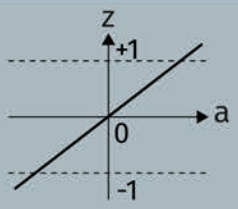
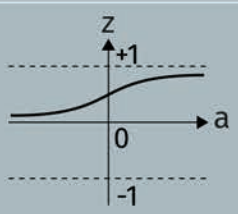
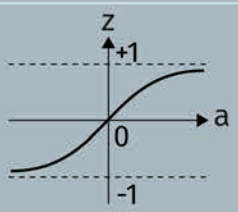
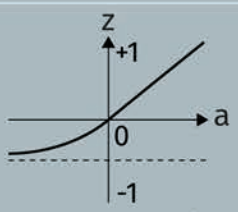
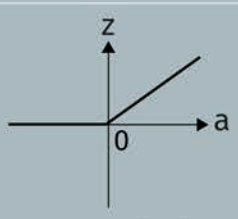
The strength of a link between two neurons (j and i) on two adjacent layers is represented by a weight value (w_{ji}). The network adjusts the weights of its links to produce an output value close to the desired value of the target variable.

The neuron sums its weighted inputs coming from its preceding links to get (a_i) and then applies a transfer function to produce output (z_i).



The transfer function (f) of a neuron is also called an activation function and should be continuous, differentiable, non-decreasing, and easy to compute. The neurons have mostly nonlinear activation functions which allow the network to learn nonlinear and linear relationships between the variables.

Typical Activation Functions

Activation function	Equation	Description
Linear (lin)	 $z = a$	The lin function generates outputs which are not confined to a specific range.
log-sigmoid (logsig)	 $z = \frac{1}{1 + e^{-a}}$	The logsig function generates outputs between 0 and 1.
tan-sigmoid (tansig)	 $z = \tanh(a) = \frac{e^{2a} - 1}{e^{2a} + 1}$	The tansig function generates outputs between -1 and +1.
Exponential linear unit (ELU)	 $z = \begin{cases} a & a > 0 \\ \alpha (e^{a} - 1) & a \leq 0 \end{cases}$ where α is a positive value, normally equals 0.01	The ELU function generates outputs which are not confined to a specific range. It has a linear shape for positive inputs and an exponential shape for negative outputs.
Rectified linear unit (ReLU)	 $z = \max(a,0)$	The ReLU function generates outputs which are 0 for inputs with negative values. For all other inputs, the output will equal the input number.

Feedforward Networks

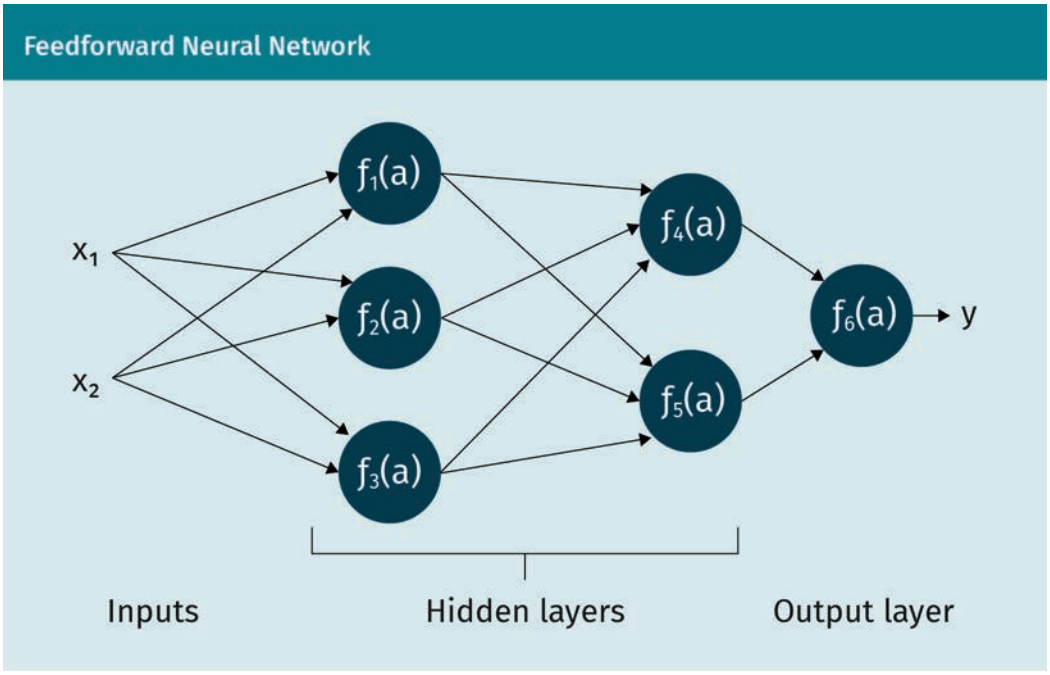
In feedforward neural networks, the neurons of each layer feedforward their output to the next layer and so forth until the outer layer of the network is reached. The number of neurons, number of hidden layers, and the neurons' activation functions are completely arbitrary (Malhotra, 2018).

As a rule, feedforward networks

1. have no connections within the neurons of a layer.
2. have no direct connections between the input layer and output layer.
3. are fully connected between layers (i.e., each neuron in one layer is connected to all neurons in its succeeding layer).
4. can have a number of hidden neurons per layer that is more or less than the number of inputs.

After specifying the number of hidden layers, number of neurons within each hidden layer, and the activation function of each neuron, the network is ready to use the given dataset to self-learn. The dataset is divided into training and testing sets. The training set (about 60 percent of the given dataset) is for learning the network, and the testing set (the remaining 40 percent) is for evaluating network performance (Haykin, 2001).

If the input dataset contains two variables (x_1 and x_2) and a target variable (d), the feedforward neural network can be developed.

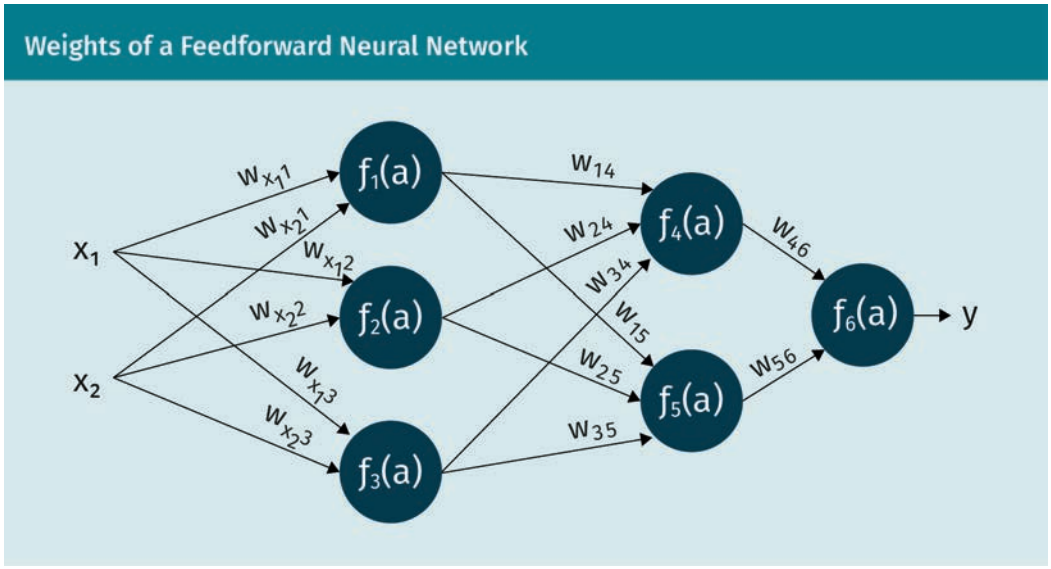


The network is composed of two hidden layers, the first of which has three neurons and the second of which has two neurons. The objective of learning the network is to get the network's output value (y) very close to the desired output (d). A common learning algorithm is the back propagation algorithm.

Back Propagation Algorithm

For multilayer, feedforward neural networks, the back propagation algorithm is used to estimate the network's weights and develop a network model which estimates an output as accurately as possible with respect to the given desired output. The desired output can be a variable value for regression problems or a class value (e.g., +1 or –1) for classification problems.

To illustrate the back propagation algorithm, we use the ANN model from the above example, with the weights and functions shown in the following figure.



For each neuron, we calculate (a) to be the sum of its weighted inputs and (z) to be its output after applying the defined transfer function (f) on (a),

$$a_i = \sum_{j=0}^{M} \left(w_{ji} z_j \right)$$

$$z_j = f\left(a_j \right)$$

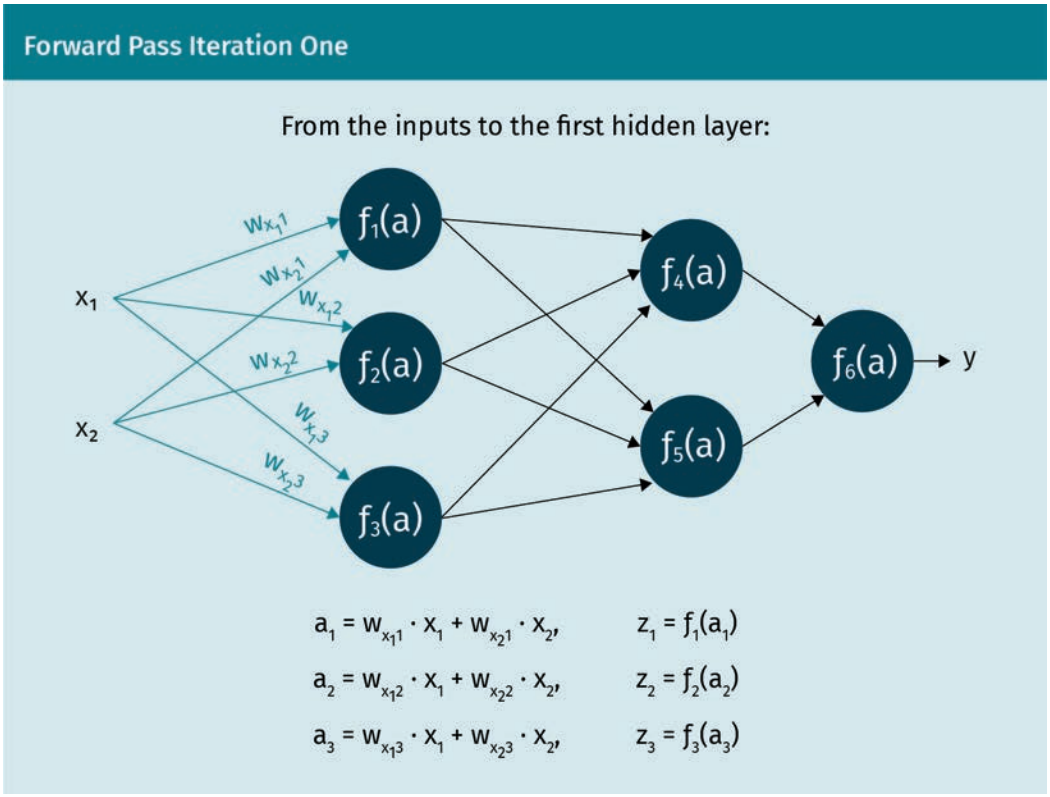
where M is the number of neurons in layer number (j).

The back propagation algorithm consists of two phases: the forward pass and the backward pass.

Forward pass phase
In the forward pass phase, the algorithm initially assumes random values for the weights and calculates all the network's parameters. The calculations proceed in a forward direction from neuron to neuron, layer to layer, all the way up to the network output (y).

The application of this phase on our ANN model is described below.

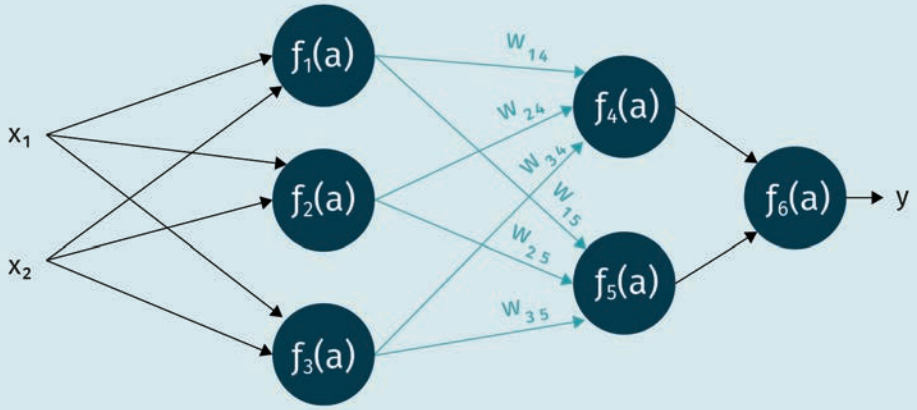
From the inputs to the first hidden layer:



From the first hidden layer to the second hidden layer:

Forward Pass Iteration Two

From the first hidden layer to the second hidden layer:



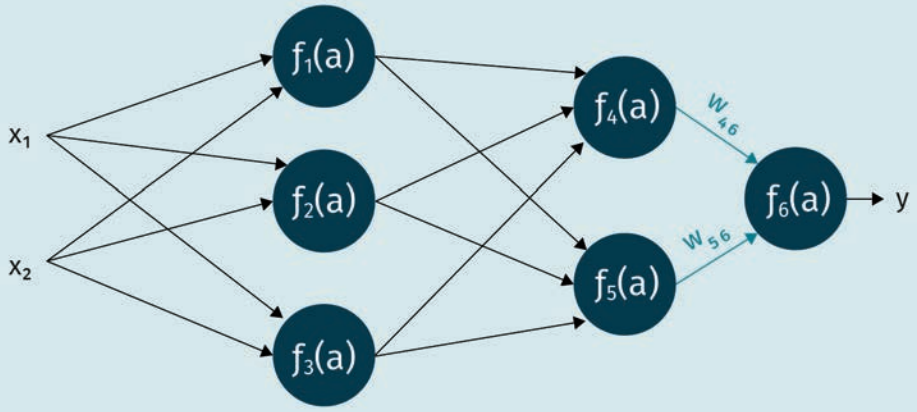
$$a_4 = w_{14} \cdot z_1 + w_{24} \cdot z_2 + w_{34} \cdot z_3, \qquad z_4 = f_4(a_4)$$

$$a_5 = w_{15} \cdot z_1 + w_{25} \cdot z_2 + w_{35} \cdot z_3, \qquad z_5 = f_5(a_5)$$

From the second hidden layer to the output layer:

Forward Pass Iteration Three

From the second hidden layer to the output layer:



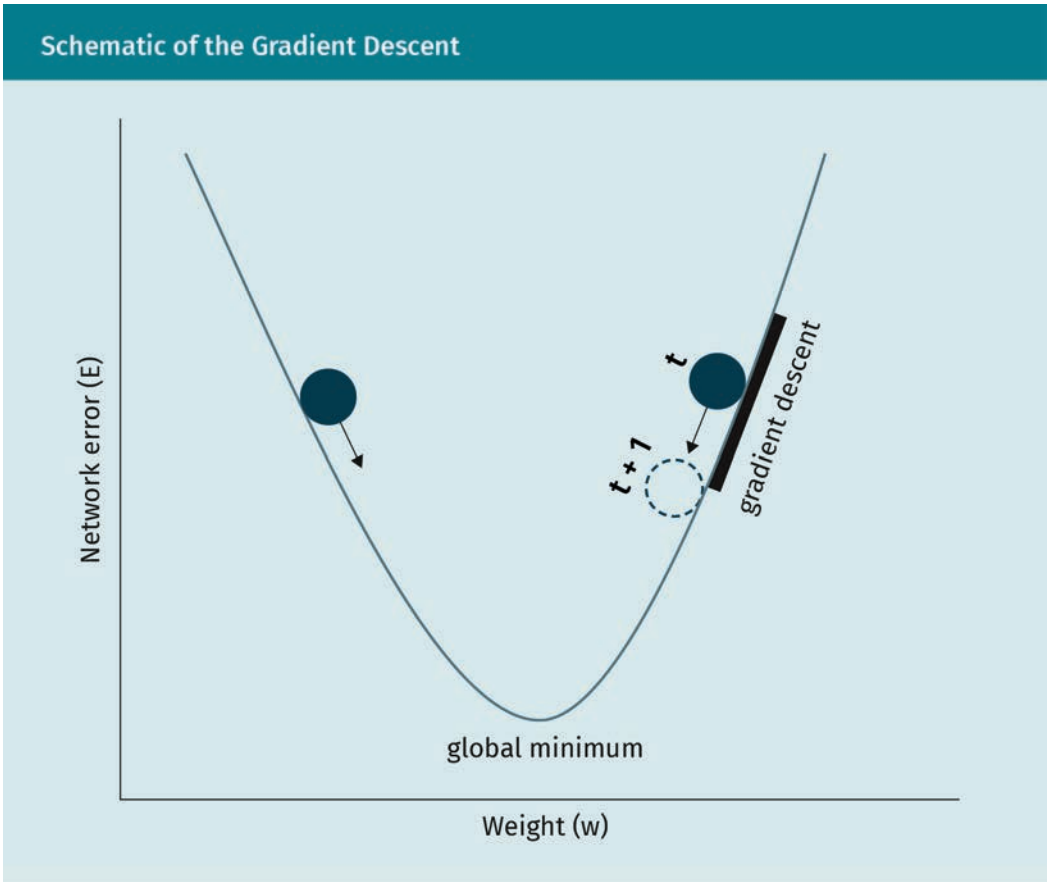
$$a_6 = w_{46} \cdot z_4 + w_{56} \cdot z_5, \qquad y_5 = z_6 = f_6(a_6)$$

The output of the network (y), with the current set of weights, is compared to the desired target value (d) in order to obtain the network error (E). The network's weights are then updated such that this error metric is decreased in the next forward pass.

Gradient descent

The "trick" of the back propagation algorithm is to use the current error metric and recursively calculate (using gradient descent) the needed adjustments to the network weights. Since there is no analytical formula to do this in one step for all weights, we work backwards through the layers and calculate how the weights need to be adjusted in order to achieve a reduced error value. After this is completed, another forward pass is performed, the resultant error is calculated, and backward pass computations are made to further adjust the network weights. The whole process is repeated until there are no more improvements in the calculated error.

The gradient descent concept acts like gravity, as schematically shown in the following figure.



In the figure, the ball moves from its current weight (w_t) to its adjusted weight (w_{t+1}) in the direction of the global minimum of the overall network error (E). It does not matter where the initial weight is because the ball will reach its stable position at the global

minimum after several adjustments. Mathematically, we see a proportional relationship between a change in the weight (Δw) and the rate of decrease in the error ($\frac{\partial E}{\partial w}$), as given in the following equations:

$$\Delta w = -\eta\left(\frac{\partial E}{\partial w}\right)$$

$$w_{t+1} - w_t = -\eta\left(\frac{\partial E}{\partial w}\right)$$

$$w_{t+1} = w_t - \eta\left(\frac{\partial E}{\partial w}\right)$$

where η is the proportional constant, called the learning rate. For real-life problems, we would not update the weights in such big steps because doing so could cause nonlinearity issues. Therefore, the value of (η) is usually in the range [0 to 1], and in many applications a value of 0.1 is used.

The overall network error (E) is defined by the mean square error between the network output (y) and the desired output (d) and averaged over the training data records [$1 \ldots n$] used to develop the model, as given in the following equation:

$$E = \frac{1}{2}\sum_{1}^{n}(d-y)^2$$

The $\frac{1}{2}$ value is included so that the power is canceled during the upcoming differentiation step. Hence, to get $\left(\frac{\partial E}{\partial w_{ji}}\right)$ at neuron (i) having a link from a preceding neuron (j), we use the chain rule:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial z_i} \cdot \frac{\partial z_i}{\partial a_i} \cdot \frac{\partial a_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial z_i} \cdot \frac{\partial f(a_i)}{\partial a_i} \cdot \frac{\partial\left(\sum_{i=0}^{M}\left(w_{ji}z_j\right)\right)}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial z_i} \cdot \frac{\partial f(a_i)}{\partial a_i} \cdot z_j$$

If the neuron is in the output layer, then:

$$\frac{\partial E}{\partial z_i} = \frac{\partial}{\partial z_i}\left[\frac{1}{2}\sum_1^n (d-y)^2\right] = \frac{\partial}{\partial z_i}\left[\frac{1}{2}\sum_1^n (d-z_i)^2\right] = -(d-z_i)$$

$$\therefore \frac{\partial E}{\partial w_{ji}} = -(d-z_i)\cdot\frac{\partial f(a_i)}{\partial a_i}\cdot z_j$$

$$\frac{\partial E}{\partial w_{ji}} = -\delta_i \cdot z_j$$

where $\delta_i = (d-z_i)\cdot\frac{\partial f(a_i)}{\partial a_i}$. Therefore, the weights of the links to the output layer will be adjusted according to the following formula:

$$(w_{ji})_{t+1} = (w_{ji})_t + \eta\cdot\delta_i\cdot z_i$$

and

$$\delta_i = (d-z_i)\cdot\frac{\partial f(a_i)}{\partial a_i}$$

Meanwhile, if the neuron (i) is in the latest hidden layer, with links j from its preceding layer and links u to its succeeding layer (i.e., the output layer):

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial z_i}\cdot\frac{\partial z_i}{\partial a_i}\cdot\frac{\partial a_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial w_{ji}} = \left(\sum_{\text{succeedinglayer neurons (u)}}\left[\frac{\partial E}{\partial z_u}\cdot\frac{\partial z_u}{\partial a_u}\cdot\frac{\partial a_u}{\partial z_i}\right]\right)\cdot\frac{\partial z_i}{\partial a_i}\cdot\frac{\partial a_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial w_{ji}} = \left(\sum_{\text{succeedinglayer neurons (u)}}\left[\frac{\partial E}{\partial z_u}\cdot\frac{\partial f(a_u)}{\partial a_u}\cdot w_{iu}\right]\right)\cdot\frac{\partial f(a_i)}{\partial a_i}\cdot z_j$$

$$\frac{\partial E}{\partial w_{ji}} = \left(\sum_{\text{succeedinglayer neurons (u)}}\left[-(d-z_u)\cdot\frac{\partial f(a_u)}{\partial a_u}\cdot w_{iu}\right]\right)\cdot\frac{\partial f(a_i)}{\partial a_i}\cdot z_j$$

$$\frac{\partial E}{\partial w_{ji}} = \left(\sum_{\text{succeedinglayer neurons (u)}}\left[-\delta_u\cdot w_{iu}\right]\right)\cdot\frac{\partial f(a_i)}{\partial a_i}\cdot z_j$$

$$\therefore \frac{\partial E}{\partial w_{ji}} = -\delta_i\cdot z_j$$

where

$$\delta_i = \left(\sum_{\text{succeedinglayer neurons (u)}} [\delta_u \cdot w_{iu}] \right) \cdot \frac{\partial f(a_i)}{\partial a_i}$$

Therefore, the weights of the links going to this hidden layer will be adjusted according to the following formula:

$$(w_{ji})_{t+1} = (w_{ji})_t + \eta \cdot \delta_i \cdot z_j$$

and

$$\delta_i = \left(\sum_{\text{succeedinglayer neurons (u)}} [\delta_u \cdot w_{iu}] \right) \cdot \frac{\partial f(a_i)}{\partial a_i}$$

Backward pass phase

In the backward pass phase of the back propagation algorithm, the process moves backward from the output layer and calculates the adjusted weights of the network.

The backward pass phase is conducted as follows:

1. Select a value for the learning rate (e.g., $\eta = 0.1$).
2. For the link connecting the latest hidden layer's neuron (j) to the output layer's neuron (i):
   - Calculate δ_i:

$$\delta_i = (d - z_i) \cdot \frac{\partial f(a_i)}{\partial a_i}$$

   - Calculate the new weight of this link:

$$(w_{ji})_{t+1} = (w_{ji})_t + \eta \cdot \delta_i \cdot z_j$$

   - Repeat the above step for all links from this hidden layer to the output layer.
3. For the link between the latest hidden layer's neuron (i) and its preceding layer's neuron (j):
   - Calculate δ_i:

$$\delta_i = \left(\sum_{\text{succeeding layer neurons (u)}} [\delta_u \cdot w_{iu}] \right) \cdot \frac{\partial f(a_i)}{\partial a_i}$$

- Calculate the new weight of this link:

$$(w_{ji})_{t+1} = (w_{ji})_t + \eta \cdot \delta_i \cdot z_j$$

- Repeat the above step for all links connected to this hidden layer from its preceding layer.

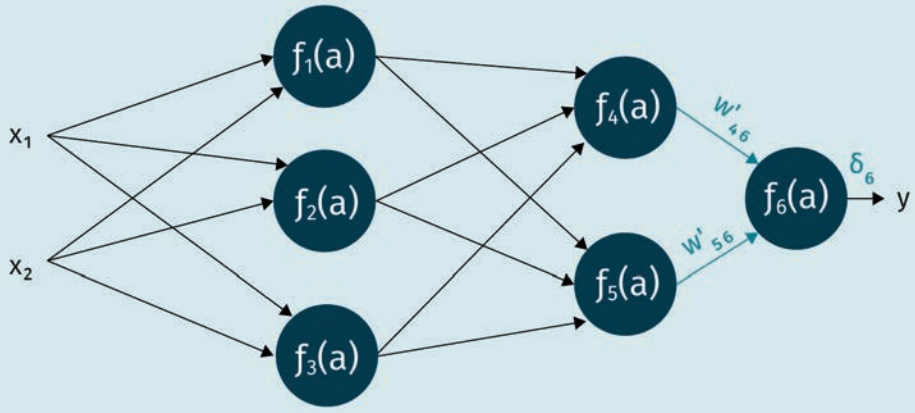
4. Repeat step (3) in a backward direction for the links between each remaining hidden layer's neuron (i) and its preceding layer's neuron (j) until the inputs' layer is reached.

The application of this phase on our ANN model is described below.

From the output layer back to the second hidden layer:

Backward Pass Iteration One

From the output layer back to the second hidden layer:



$$\delta_6 = (d - z_6) \cdot \frac{\partial f_6(a_6)}{\partial a_6}$$

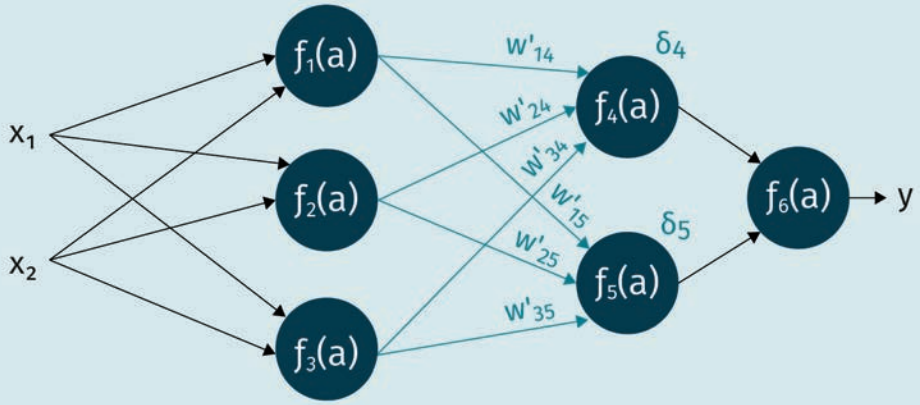
$$w'_{46} = w_{46} + \eta \cdot \delta_6 \cdot z_4$$

$$w'_{56} = w_{56} + \eta \cdot \delta_6 \cdot z_5$$

From the second hidden layer back to the first hidden layer:

Backward Pass Iteration Two

From the second hidden layer back to the first hidden layer:



$$\delta_4 = [\delta_6 \cdot w_{46}] \cdot \frac{2f_4(a_4)}{2a_4}$$

$$\delta_5 = [\delta_6 \cdot w_{56}] \cdot \frac{2f_5(a_5)}{2a_5}$$

$$w'_{14} = w_{14} + \eta \cdot \delta_4 \cdot z_1, \quad w'_{15} = w_{15} + \eta \cdot \delta_5 \cdot z_1$$

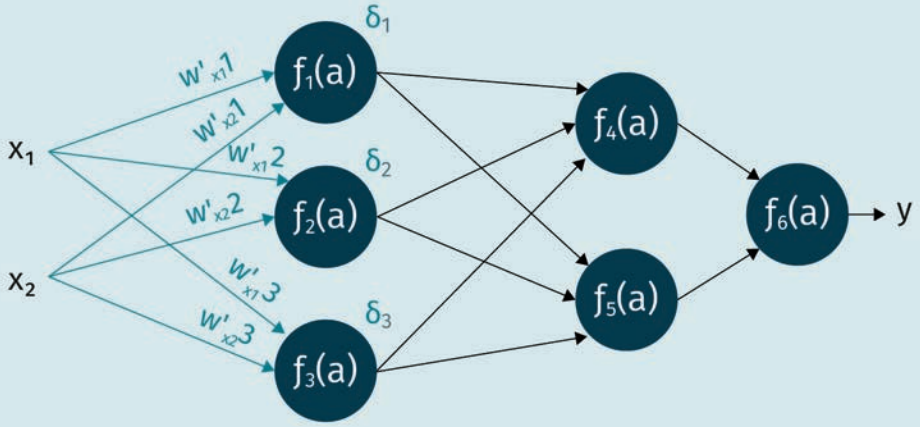
$$w'_{24} = w_{24} + \eta \cdot \delta_4 \cdot z_2, \quad w'_{25} = w_{25} + \eta \cdot \delta_5 \cdot z_2$$

$$w'_{34} = w_{34} + \eta \cdot \delta_4 \cdot z_3, \quad w'_{35} = w_{35} + \eta \cdot \delta_5 \cdot z_3$$

From the first hidden layer back to the inputs' layer:

Backward Pass Iteration Three

From the first hidden layer back to the inputs' layer:



$$\delta_1 = [\delta_4 \cdot w_{14} + \delta_5 \cdot w_{15}] \cdot \frac{2f_1(a_1)}{2a_1}$$

$$\delta_2 = [\delta_4 \cdot w_{24} + \delta_5 \cdot w_{25}] \cdot \frac{2f_2(a_2)}{2a_2}$$

$$\delta_3 = [\delta_4 \cdot w_{34} + \delta_5 \cdot w_{35}] \cdot \frac{2f_3(a_3)}{2a_3}$$

$$w'_{x_1 1} = w_{x_1 1} + \eta \cdot \delta_1 \cdot x_1, \quad w'_{x_1 2} = w_{x_1 2} + \eta \cdot \delta_2 \cdot x_1, \quad w'_{x_1 3} = w_{x_1 3} + \eta \cdot \delta_3 \cdot x_1$$

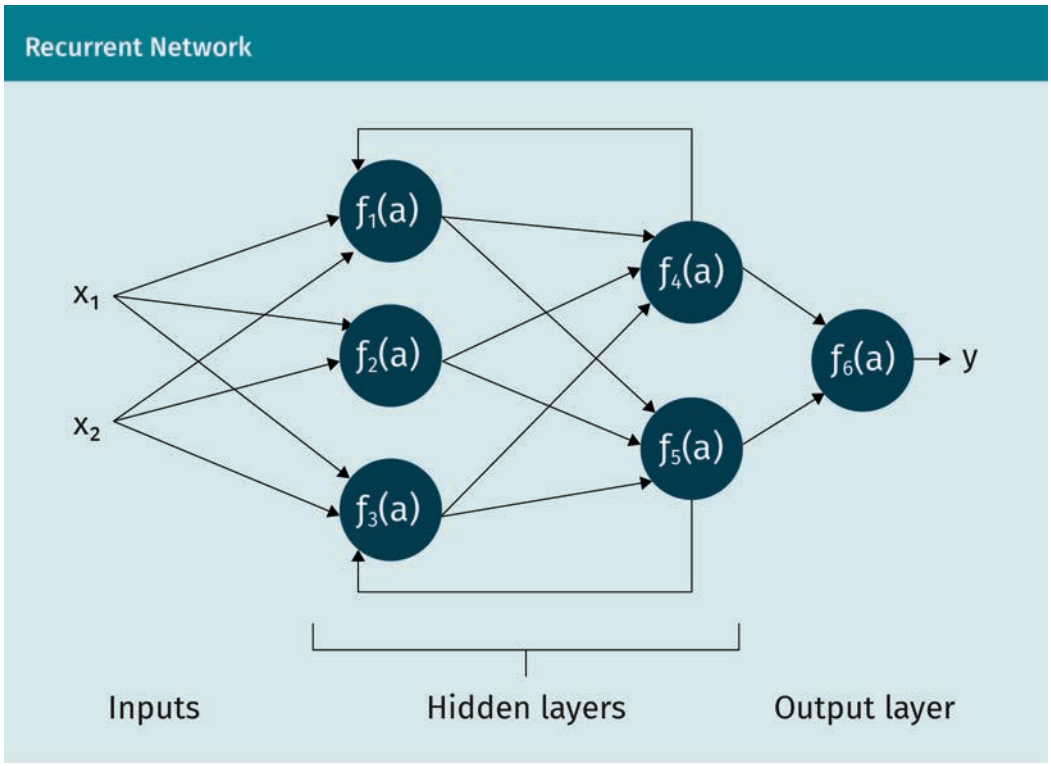
$$w'_{x_2 1} = w_{x_2 1} + \eta \cdot \delta_1 \cdot x_2, \quad w'_{x_2 2} = w_{x_2 2} + \eta \cdot \delta_2 \cdot x_2, \quad w'_{x_2 3} = w_{x_2 3} + \eta \cdot \delta_3 \cdot x_2$$

The above process is for a single iteration of the back propagation algorithm, where the weights are adjusted only once. After each iteration, the network error (E) is re-calculated for the testing set, and it is checked whether it reached its global minimum value or a reasonable low value. If the value of (E) is still high, we repeat the iteration until there is no additional improvement in the calculated network error.

If the network's error cannot be further improved and it neither reached its goal minimum value nor an acceptable value, we may consider changing the implemented neurons' activation functions.

Recurrent Networks and Memory Cells

The human brain operates in a more complex manner than a feedforward network, but it permits links to occur from succeeding layers to preceding layers (i.e., feedback). In ANN, this model is called a recurrent network.



The idea of recurrent networks is to allow connections to previous layers and to some neurons in the same layer. Recurrent networks can also be trained using the back propagation algorithm; this algorithm needs to remember the previously-obtained values of the recurrent connections. This is the concept of a memory cell, which allows new terms to be added into the network's mathematical functions. It influences what, and to what degree, the output of a recurrent neuron is fed back to specific neurons. An example of these memory cells is the long short-term memory (LSTM) unit, comprised of input, output, and forget gates. LSTM remembers values from the recurrent neurons over arbitrary time intervals, and the three gates regulate the flow of this information through the network.

In the learning and training of recurrent networks, it may be a long time before a stable output with minimum overall error is reached. In the above figure, the outputs of neurons (4) and (5) are fed back as inputs to their previous layer's neurons (1) and (3), respectively. This means that from one step to the next, each neuron has to remember information from the previous step. Therefore, these neurons act as memory cells capable of performing the required computations.

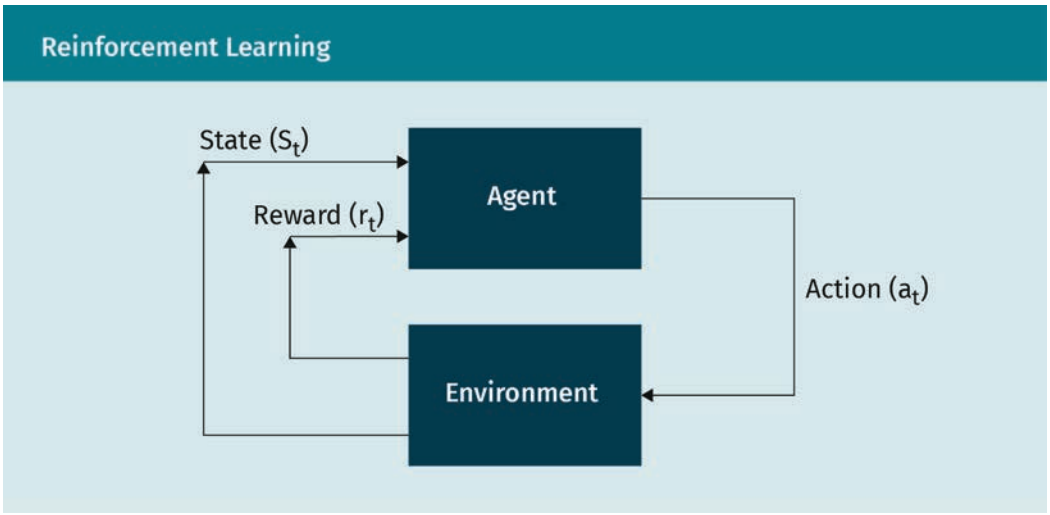
$$(a_1)_{t+1} = \left(w_{x_1 1} \cdot x_1 + w_{x_2 1} \cdot x_2\right)_{t+1} + (w_{41} \cdot z_4)_t$$

$$(a_3)_{t+1} = \left(w_{x_1 3} \cdot x_1 + w_{x_2 3} \cdot x_2\right)_{t+1} + (w_{53} \cdot z_5)_t$$

During the forward pass phase of the applied back propagation algorithm, the recurrent network memorizes what information it needs for the next iterations. The backward phase of the back propagation algorithm proceeds according to the gradient descent criteria.

Reinforcement Learning

Reinforcement learning is a goal-oriented learning approach based on interaction with the environment and investigates all possible scenarios to find the optimal correct actions. The objective of applying reinforcement learning is to allow the learner (the machine) to notice which action yields the maximum reward. The classic structure of a reinforcement learning problem is shown below, which consists of an artificial agent with a feedback loop to reinforce the agent. It rewards the agent when the performed actions are correct (Shaikh, 2017).



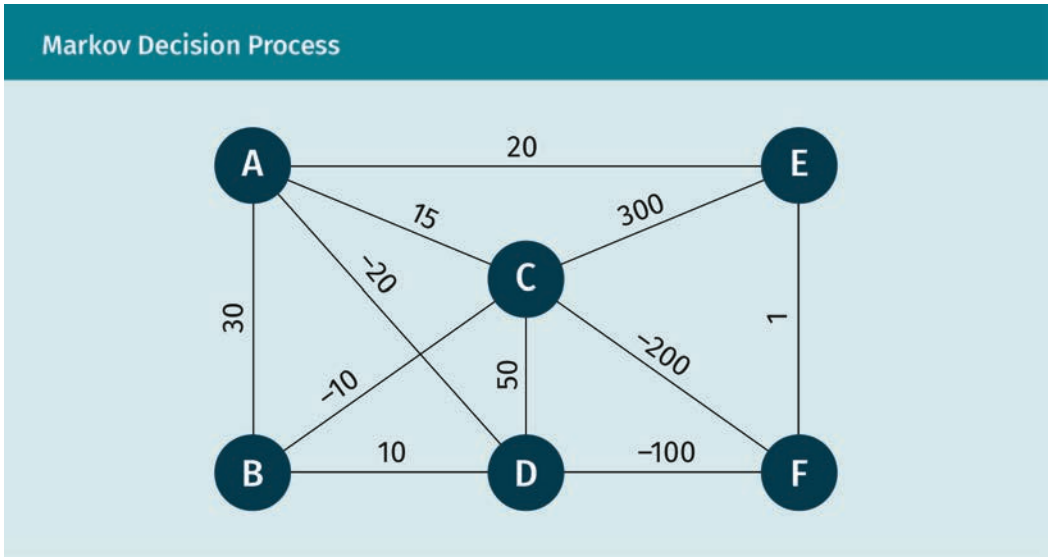
The environment is the scenario the agent is facing, and the internal state of the scenario is maintained by the agent in order to learn about the environment. Hence, the agent performs an action within the environment, and a reward function is used to train the agent how to behave.

Comparison of learning types: supervised, unsupervised, reinforcement
Both supervised learning and reinforcement learning map from inputs to outputs in their developed models, but with reinforcement learning, a reward function acts as feedback to the agent. This is different from supervised learning which predicts the mathematical function that relates the outputs to the given inputs. With unsupervised learning there is no mapping from inputs to outputs, but the objective of the learning algorithm is to find underlying patterns in the dataset.

Markov decision process

The Markov decision process is the mathematical framework for solving reinforcement learning problems. The process consists of the main parameters for reinforcement learning: {S, A, R, P, and V}, which stand for states, actions, reward, policy, and value, respectively. The policy (P) is the set of actions the agent will take, and the value (V) is the total reward achieved by following this policy.

A reinforcement learning example which applies the Markov decision process is the shortest path problem given below.



The objective of this shortest path problem is for the agent to go from place (A) to place (F) at the lowest cost. The weight of each path defines the cost of traversing a specific path. The parameters are

- set of states {A, B, C, D, E, F}
- set of actions to take a specific path (e.g., {A→C})
- set of rewards, which are the weights of the paths (i.e., cost)
- the policy, which is the selected path to complete the task (e.g., {A→E→F})

Starting from place {A}, the only visible paths are those to possible destinations {B, C, D, and E}; any progression before place {A} is not visible. Therefore, the path with the lowest cost is {A→D}. Similarly, to move forward from place {D}, the best path will be {D→F}. Thus the policy is {A→D→F}, with the value of total reward equal to ({–20} + {–100} = –120).

In the solution to the above reinforcement learning problem, there were three categories to use as bases: policy-based (to solve for the optimal policy), value-based (to solve for the optimal value), and action-based (to solve for the optimal actions).

A common algorithm for policy-based reinforcement learning can be summarized in the following steps.

1. Initialize the parameters $\{S, A, R, P, \text{and } V\}$.
2. Observe the current state $\{S_t\}$.
3. Choose an action $\{r_{t+1}\}$ according to the maximum possible reward for the next state.
4. Take the action and reach the new state $\{S_{t+1}\}$.
5. Update the value $\{V\}$.
6. Repeat the process until the terminated (end) state is reached.

6.3 Further Approaches

Genetic Algorithm

The genetic algorithm (GA) is an approach to solving optimization problems based on natural selection. The GA starts by generating a population of possible solutions and then applies selection rules to randomly select individuals from the current population to be parents. The crossover rules are applied to combine two parents and form children for the next generation. Mutation rules are applied with random changes to the parents to form different children. Over successive generations, the population evolves toward the optimal solution (in this case, children with the best genetic combinations).

Fuzzy Logic

Fuzzy logic is almost synonymous with the theory of fuzzy sets which deals with classes of objects with unsharp boundaries. The membership of these clusters is based on degrees of truth instead of the usual {+1, –1} assignments. The first step of fuzzy logic is to fuzzify (decompose) all input values into truth values, which are any real numbers between “0” (completely false) and “1” (completely true). The fuzzy output is computed by the execution of a group of “IF-THEN” rules in a way that mimics Boolean logic operators (“AND”, “OR”, and “NOT”). Finally, a de-fuzzification step is performed on the fuzzy truth values to get a continuous output value.

Naïve Bayes Classification

The Naïve Bayes method is a classification approach based on Bayes theorem and designed for categorical data. In this approach, a new data record is assigned to the class which returns the highest probability that this data record belongs to it. The Naïve Bayes approach assumes that the independent variables are random variables; this way, they can be utilized to calculate the required probabilities even if they contain very few values.

Summary

The support vector machines approach is used to develop a classification prediction model. The model builds a separating channel between the data classes, where the support vectors are the data records which lie on either side of the developed channel. The approach runs smoothly to classify linearly-separable data and requires use of the Kernel trick if the underlying data is nonlinearly separable. The Kernel trick adopts a suitable Kernel function which projects the data records into a higher domain where they become linearly separable. Hence, the support vector machines technique tries to define the separating hyperplane which maximizes the margin between the different classes of the input data.

For nonlinear regression problems, artificial neural networks are often used because they mimic the biological networks of the human brain. The feedforward network is composed of neurons distributed on connected layers from the input independent variables to the output target variable. Each neuron sums up its weighted inputs and applies an activation function to the result before presenting this result to the succeeding layer. To learn the neural network model, the training set of the given dataset is utilized. A common learning algorithm is the back propagation algorithm.

The recurrent network works similar to the feedforward neural network but in a more complicated manner because it contains feedback connections between some neurons. Reinforcement learning—similar to supervised learning—maps from the inputs to the outputs in the developed models, but with reinforcement learning a reward function acts as feedback to the learner. Finally, there are many other approaches which are implemented for analysis and making predictions in the data science field. Examples of these approaches are fuzzy logic, genetic algorithm, and the Naïve Bayes classifier.

Knowledge Check

Did you understand this unit?

Now you have the chance to test what you have learned on our Learning Platform.

Good luck!

Congratulations!

You have now completed the course. After you have completed the knowledge tests on the learning platform, please carry out the evaluation for this course. You will then be eligible to complete your final assessment. Good luck!

Appendix 1

List of References

List of References

Baldassarre, M. (2016). Think big: Learning contexts, algorithms and data science. *Research on Education and Media, 8*(2), 69–83. Retrieved from https://content.sciendo.com/view/journals/rem/8/2/article-p69.xml

Brownlee, J. (2017, January 9). How to create an ARIMA model for time-series forecasting in Python [blog post]. Retrieved from https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: Library for support vector machines [guide and software download]. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Retrieved from https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Dalinina, R. (2017, January 10). Introduction to forecasting with ARIMA in R [blog post]. Retrieved from https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials

Das, D., & Sharma, B. (2016). General survey on security issues on Internet of Things. *International Journal of Computer Applications, 139*(2), 23–29.

Das, S. R. (2017). Data science: Theories, models, algorithms, and analytics [paper]. Retrieved from https://srdas.github.io/MLBook/

Datameer, Inc. (2016). Top 5 high-impact use cases for big data analytics [downloadable white paper]. Retrieved from https://www.datameer.com/pdf/eBook-Top-Five-High-Impact-UseCases-for-Big-Data-Analytics.pdf

Doll, J., & Jacquemin, S. J. (2018). Bayesian inference for fisheries scientists. *Fisheries Magazine, 43*(3), 152–161.

Dorard, L. (2017). The machine learning canvas [PDF document]. Retrieved from https://www.louisdorard.com/machine-learning-canvas

Hackernoon. (2018, June 2). General vs narrow AI [blog post]. Retrieved from https://hackernoon.com/general-vs-narrow-ai-3d0d02ef3e28

Haykin, S. (2001). Feedforward neural networks: An introduction. In I. W. Sandberg, J. T. Lo, C. L. Fancourt, J. C. Principe, S. Katagiri, & S. Haykin (Eds.), *Nonlinear Dynamical Systems* (pp. 1–14). New York, NY: Wiley.

Helmenstine, A. (2017, March 16). Bayes theorem definition and examples [blog post]. Retrieved from https://www.thoughtco.com/bayes-theorem-4155845

Internet Movie Database (IMDB). (n.d.). Mark Wahlberg [entry]. Retrieved from https://www.imdb.com/name/nm0000242/

List of References

Le Dem, J. (2016). Efficient data formats for analytics with Parquet and Arrow [presentation slides]. Retrieved from https://2016.berlinbuzzwords.de/sites/2016.berlinbuzzwords.de/files/media/documents/berlin_buzzwords_2016_parquet_arrow.pdf

Lucid Software. (n.d.). Lucidchart [diagramming app]. Retrieved from https://www.lucidchart.com

Make Knowledge Free. (2013, July 28). How to generate XML file from ITR tax return [video slide]. Retrieved from https://www.youtube.com/watch?v=0SdfLu9emGE

Malhotra, A. (2018). Tutorial on feedforward neural network—Part 1 [tutorial]. Retrieved from https://medium.com/@akankshamalhotra24/tutorial-on-feedforward-neural-network-part-1-659eeff574c3

Mamchenkov, L. (2013). Cognitive bias codex [illustration]. Retrieved from http://mamchenkov.net/wordpress/2017/06/13/list-of-cognitive-biases/

Montibeller, G., & Winterfeldt, D. (2015). Cognitive and motivational biases in decision and risk analysis. *Risk Analysis, 35*(7), 1230–1251.

Nau, R. (2014). Notes on nonseasonal ARIMA models [PDF document]. Retrieved from http://people.duke.edu/~rnau/Notes_on_nonseasonal_ARIMA_models--Robert_Nau.pdf

Pathak, M. (2014). Regression. In *Beginning data science with R* (pp. 87–114). Cham: Springer.

PeerXP. (2017, October 17). The 6 stages of data processing cycle [blog post]. Retrieved from https://medium.com/peerxp/the-6-stages-of-data-processing-cycle-3c2927c466ff

Polson, N., & Scott, S. (2011). Data augmentation for support vector machines. *Bayesian Analysis, 6*(1), 1–23. Retrieved from https://projecteuclid.org/euclid.ba/1339611936#

Prakash, R. (2018, June 19). 5 different types of data processing [video]. Retrieved from https://www.loginworks.com/blogs/5-different-types-of-data-processing/

Rapolu, B. (2016, January 18). Internet of aircraft things: An industry set to be transformed [article]. Retrieved from http://aviationweek.com/connected-aerospace/internet-aircraft-things-industry-set-be-transformed

Robinson, J. (2018, May 14). What does your favourite player's heatmap say about them? [article with graphic]. *Dream Team*. Retrieved from https://www.dreamteamfc.com/c/news-gossip/397578/favourite-player-heatmap-lionel-messi/

Runkler, T. A. (2012). *Data analytics: Models and algorithms for intelligent data analysis*. Wiesbaden: Springer Vieweg.

Shaikh, F. (2017, January 19). Simple beginner's guide to reinforcement learning & its implementation [blog post]. Retrieved from https://www.analyticsvidhya.com/blog/2017/01/introduction-to-reinforcement-learning-implementation/

Skiena, S. S. (2017). *The data science design manual*. Cham: Springer.

Statista. (2019, August 9). Hours of video uploaded to YouTube every minute as of May 2019 [article]. Retrieved from https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/

Thakur, D. (2017). What is data transmission? Types of data transmission [article]. Retrieved from http://ecomputernotes.com/computernetworkingnotes/communication-networks/data-transmission

TheAILearner. (2019). Power law (gamma) transformations [article, graphics]. Retrieved from https://theailearner.com/2019/01/26/power-law-gamma-transformations/

Tierney, B. (2012, June 13). Data science is multidisciplinary [blog post]. Retrieved from https://www.oralytics.com/2012/06/data-science-is-multidisciplinary.html

U.S. Energy Information Administration (EIA). (2018, December 12). Weekly net crude oil and petroleum product trade, Jan 2000–Nov 2018 [graph]. Retrieved from https://www.eia.gov/todayinenergy/detail.php?id=37772

Wenzel, F., Galy-Fajou, T., Deutsch, M., & Kloft, M. (2017, September 18). Bayesian nonlinear support vector machines for big data. Presented at the *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases*, Skopje. Retrieved from https://arxiv.org/abs/1707.05532

Wikipedia. (n.d.). Data transformation [entry]. Retrieved from https://en.wikipedia.org/wiki/Data_transformation_(statistics)

Appendix 2

List of Tables and Figures

List of Tables and Figures

IMDB Actor Entry
Source: IMDb, n.d.

Venn Diagram for Data Science
Source: Tierney, 2012.

Bayesian Statistics
Source: Author, based on Doll & Jacquemin, 2018.

Drug Testing Analysis Using Bayesian Statistics
Source: Helmenstine, 2017.

Value Propositions in Customer-Related DSUCs
Source: Author, based on Datameer, 2016.

Value Propositions in Operational-Related DSUCs
Source: Author, based on Datameer, 2016.

Value Propositions in Fraud-Related DSUCs
Source: Author, based on Datameer, 2016.

Machine Learning Canvas
Source: Dorard, 2017.

Using the Machine Learning Canvas for Real Estate Deals
Source: Author, based on Dorard, 2017.

Cognitive Bias
Source: Mamchenkov, 2013.

Common Cognitive and Motivational Biases
Source: Author, based on Montibeller & Winterfeldt, 2015.

De-biasing Techniques
Source: Author, based on Montibeller & Winterfeldt, 2015.

List of Tables and Figures

Bubble Chart
Source: Lucid Software, n.d.

Heat Map
Source: Robinson, 2018.

Extensible Markup Language Format
Source: Author, based on Make Knowledge Free, 2013.

Apache Parquet Format
Source: Le Dem, 2016.

Time-series Data Example: U.S. Petroleum Products Trade
Source: U.S. Energy Information Administration, 2018.

Example (1): ARIMA (2, 0, 0) Model
Source: Author, based on Nau, 2014.

Example (2): ARIMA (0, 0, 1) Model
Source: Author, based on Nau, 2014.

Example (3): ARIMA (2, 1, 5) Model
Source: Author, based on Nau, 2014.

Example (3): ARIMA (2, 1, 5) Model (ACF/PACF)
Source: Author, based on Nau, 2014.

Logarithm Transformation
Source: Author, based on Wikipedia, n.d.

The Power Law Transformation
Source: Author, based on TheAILearner, 2019.

Support Vector Machines
Source: Author, based on Pathak, 2014.

All other tables and figures
Source: Author.

IU Internationale Hochschule GmbH
IU International University of Applied Sciences
Juri-Gagarin-Ring 152
D-99084 Erfurt

Mailing address:
Albert-Proeller-Straße 15-19
D-86675 Buchdorf

Phone: +49 30 311 988 55
media@iu.org