

# CHAPTER 1

## Data Preprocessing and Feature Engineering

**Bart Baesens, Wouter Verbeke, Johannes De Smedt,  
Jochen De Weerd, Hans Weytjens**

**Corresponding author: [Bart.Baesens@kuleuven.be](mailto:Bart.Baesens@kuleuven.be)**

### 1.1 CHAPTER OBJECTIVES

In this chapter, you learn to

- understand the need for data preprocessing and identify different types of data and variables;
- why and how to sample and denormalize data;
- summarize data using visual data exploration and descriptive statistics;
- identify and deal with missing values, outliers and standardize data;
- code categorical variables and do categorization of both categorical as well as continuous variables;
- calculate the weights of evidence and information value of a categorized variable;
- do feature engineering to boost the performance and/or interpretability of your analytical models;

### 1.2 INTRODUCTION

Data preprocessing and feature engineering are two key activities to develop high-performing analytical models. Real-life data is typically dirty, noisy or can have unexpected values. Examples are age is -2003 years, monthly income=1 million Euro, age is 28 years whilst at the same time customer tenure is 32 years, etc. Dirty data can have various origins. It can relate to sloppiness during data entry or due to inadequate data integration or merging. For example, different data sources can express amounts in various currencies such as Euro, US dollar, Chinese Yuan, etc. Hence, when merging these special care is needed to make sure that a consistent representation is obtained. Also socio-demographic information can be measured either in different ways (e.g., m/f/x, female/male/x for gender) or at different levels of granularity (e.g., city level versus street level for address). Another rather popular example of inconsistent data concerns the

meaning of the value 0. In certain data sources this can refer to the numerical value 0, whereas in others it can indicate a missing value. In fact, missing values can be encoded in different ways across data sets (e.g., as 'X', 0, '-', '?', etc). In other words, it is important that data is always coded in a consistent format especially when it needs to be combined for analytical model development.

Real-life data sets are typically contaminated with outliers and missing data both of which should be taken care of. Duplicate data can also occur, think about the variables, salary, income, wage and pay, which could all originate from different sources but essentially measure the same thing. Adequately preprocessing data is an important activity during the development of an analytical model. This can be motivated by the GIGO principle, which refers to "garbage in, garbage out." The meaning is that bad data leads to bad analytical models. Hence, it is important to carefully consider all the necessary preprocessing activities and take sufficient time to execute them. It is commonly known that preprocessing data uses approximately 80% of the total analytical model development effort.

The aim of feature engineering is to transform the data by enriching it using features that are defined either as transformations of existing raw data elements or based on business knowledge. Proper feature engineering can help boost the performance and/or interpretability of analytical models.

In fact, the aim of data preprocessing and feature engineering is two-fold. First, it deals with the noise and data quality issues of the data. Secondly, it optimally transforms the source data or sample so as to give the subsequent analytical model the best possible starting point to find analytically meaningful patterns in the data.

In what follows, we elaborate on key data preprocessing and feature engineering activities. We kick off the chapter by discussing types of data, sampling and types of variables in Sections 1.3, 1.4 and 1.5. We then review how to properly define variables and targets in Sections 1.6 and 1.7. Data normalization is covered in Section 1.8. Next, we discuss ways to explore your data, either visually in Section 1.9 or using descriptive statistics in Section 1.10. We then elaborate on how to deal with missing values and outliers in Sections 1.11 and 1.12. Data standardisation is covered in Section 1.13. We zoom in on categorical variable encoding in Section 1.14. This is followed by discussion of categorization, weights-of-evidence (WOE) and Information Value in Sections 1.15 and 1.16. We conclude the data preprocessing part of this chapter by elaborating on data quality in Section 1.17. Section 1.18 takes a deep dive on feature engineering. It starts by discussing its importance and then continues and zooms in on the well-known RFM features, domain specific features, trend features and transformation based features.

## 1.3 TYPES OF DATA

Different types of data can be gathered for analytical model development. In what follows, we elaborate on master data, transactional data, external data, open data, big data, structured and unstructured data, and metadata. Note that these essentially offer different perspectives of looking at data and can thus be overlapping.

Master data relates to the core entities a company is working with such as customers, products, employees, suppliers and vendors. The data involved is typically very static and uniformly defined across the various business units within an organization. Before embarking on any analytical project, it is important to properly identify all master data eligible for modeling.

Transactional data is typically gathered internally by the firm and pertains to the timing, quantity and items involved in a transaction. Think about a physical Point of Sale (PoS) application recording the items, location and timing of a transaction or on-line transaction processing systems as adopted by Amazon or Netflix. Transactional data are often summarized using the Recency, Frequency and Monetary (RFM) features variables as we discuss in Section 1.18.2.

External data is getting more and more important in contemporary analytical model development. Various examples can be thought of. Social media data obtained from LinkedIn, Facebook, Twitter, Instagram, etc.

can be used for sentiment analysis. Companies often use this to monitor brand reputation. Macro-economic data such as GDP, inflation and unemployment can be used to study the impact of stress events such as recessions on analytical models. Weather data such as temperature can be useful for forecasting soft drink sales or in smart agriculture for irrigation purposes using, e.g., IoT sensors. Competitor data such as marketing actions and discounts can be used for churn prediction. Search engine data such as Google Trends can be used for nowcasting where the aim is to forecast the present or near future. Think about forecasting unemployment based upon Google searches with key terms jobs, unemployment benefits and social security. Web scraped data can also be an interesting source of data. As an example, consider a list of reviews scraped from a movie site to perform text analytics, create a recommendation engine or build a predictive model to spot fake reviews [Baesens and vanden Broucke, 2018]. Other examples of external data are government data as provided for example by organizations such as Eurostat and OECD or dedicated database such as the Human Genome project which records human genomic sequence information and makes it publicly available to everyone. Finally, many modern day deep learning models have been pre-trained on publicly available external data. For example, convolutional neural networks such as AlexNet and the Resnet variants have been pretrained on the ImageNet database, a publicly available data source of about 14 million images that have been hand-annotated into more than 20,000 categories. Also large language model such as BERT, trained on the English Wikipedia and the Brown Corpus, and ChatGPT trained on a time snapshot of about the entire Internet are examples of this.

Modern day data is often referred to as Big data. In a 2001 research report, Gartner set out to define the scope of Big Data by listing its characteristics in the now-famous three V's: Volume (the amount of data, also referred to the data "at rest"), Velocity (the speed at which data comes in and goes out, data "in motion"), and Variety (the range of data types and sources that are used, data in its "many forms"). In recent years, vendors and researchers have also argued for a fourth V to be included in the description of Big Data: Veracity, or data "in doubt." It describes the uncertainty due to data inconsistency and incompleteness, to ambiguities present in the data, as well as latency or certain data points that might be derived from estimates or approximations. Finally, to emphasize that being able to store or process these forms of data is not enough, many vendors, such as IBM, have also included an obvious though crucial fifth V: Value. This is the end game – after spending a lot of time, effort and resources in setting up a Big Data initiative, one needs to make sure that actual value is being derived from doing so. This V refers specifically to the economic value of Big Data as quantified using the Total Cost of Ownership (TCO) and Return on Investment (ROI).

### Big Data Examples.

One example of Big Data are Rolls Royce's airplane engines which are packed with sensors generating hundreds of terabytes, which can then be analyzed to improve fleet performance and safety. Another captivating example is Tesla's Autopilot, which has so far collected more than 1 billion miles of data and is being used by the company to continuously improve its self-driving software. Finally, it is estimated that eBay.com works with a data warehouse of 40 petabytes (that's 40000 terabytes!).

Another data categorisation concerns the difference between structured and unstructured data as illustrated in Figure 1.1.

With structured data, individual characteristics of data items can be identified and formally specified, such as the number, name, address and email of a customer or the number and name of a product. With unstructured data, there are no finer grained components in a file or series of characters that can be interpreted in a meaningful way. Popular examples are text, images, audio and video. Finally, semi-structured data is data which does have a certain structure, but the structure may be very irregular or highly volatile. Typical examples are individual users' webpages on a large social media platform, or resume documents in a human resources database, which may loosely exhibit the same structure, but which do not comply entirely with a single, rigid, format. It is assumed that about 80% of all firm data is unstructured.

Metadata is data that describes other data. Examples are author of a document, date it was created,

### Structured Data

Customer	Bart Baesens
Date of Birth	February 27 <sup>th</sup> , 1975
Occupation	Professor of Data Science
Address	Naamsestraat 69, B-3000 Leuven, Belgium

### Unstructured Data

Professor Bart Baesens is a professor of Data Science at KU Leuven (Belgium), and a lecturer at the University of Southampton (United Kingdom). He has done extensive research on big data & analytics, credit risk modeling, fraud detection, and marketing analytics. He regularly tutors, advises and provides consulting support to international firms with respect to their analytics and credit risk management strategy.

Figure 1.1: Structured versus Unstructured Data.

number of words, etc. It also includes data definitions that describe the meaning of data items such as how are dates defined, in what currency are amounts expressed, etc. Metadata can turn out to be very predictive in certain AI applications. Consider the application of fraud detection. If the date and location of a picture depicting a car accident do not match the actual date and location of the accident, then a suspiciousness flag can be raised and further follow-up is needed. Another example is web analytics where metadata such as type of operating system and IP address can be used to identify web visits and do geospatial analysis.

Finally, note that analytics is all about analyzing data and obtaining insights and patterns from it, but this does not necessarily have to be applied on huge volumes or unstructured data sets. Hence, also small data sets can be successfully leveraged. Popular examples of this are very specific products (e.g., project finance in credit), luxury products or new products for which not a lot of data is available yet.

## 1.4 SAMPLING

A sample is a finite reference data set of a representative population, so as to have good generalization behaviour when applying the analytical model estimated on it to the future target population. It is true that, with the availability of high performance computing facilities (e.g., grid, cloud computing, GPUs, etc), one could also try to directly analyze the full data set. However, a key requirement for a good sample is that it should be representative for the future entities on which the analytical model will be run. In other words, customers of tomorrow are probably more similar to customers of the last year than to customers of, e.g., the pre-COVID era. A key question is how far one should go back in time when sampling.

First and foremost, the sample should be representative for the target population in terms of, e.g., sector, region, size, age and/or product composition. When the historical data contains subpopulations that are no longer of interest (e.g., due to a customer portfolio that was either sold or ceased to exist), these may be omitted in the sample. Specific subpopulations with, e.g., high-valued customers or high-risk profiles may be given more weight depending upon the application. Often, a data set covering multiple business cycles is preferred so as to capture the different characteristics of the different macro-economic upturn and downturn periods. When doing so, one needs to make a trade-off between using old term data and recent data. The latter is typically more similar to future data as mentioned earlier.

Sometimes stratified sampling is applied when one partitions the data in disjunct or mutually exclusive subpopulations according to some important criteria (e.g., fraud/default/churn rate). In the sampling step, the proportions of the strata in the sample are then aligned with the strata proportions in the source population.

Seasonal patterns can cause a bias when the moment of data collection differs from the moment of data use. Social media and payment transaction data are available almost continuously. Credit risk scores are updated almost daily. When the data exhibit seasonal trends, e.g., because of the year-end holiday season, a bias may occur when developing a model on historical year-end data and then applying it throughout the year including months with lower transaction activity. The bias can be avoided by sampling the model data throughout the whole year instead of using only year-end data. When using financial statements or accounting data, quarterly seasonality patterns may exist as well. Hence, one needs to be attentive for

potential biases when developing the model on annual year-end data and applying it on quarterly updated data during the year.

### **Reject Inference as a sampling problem in credit scoring.**

Before using any analytics or AI for credit risk modeling, banks were using credit acceptance policies based on expert intuition or common sense which were assembled during many years of business experience [Baesens et al., 2016]. For example, it is quite obvious that a customer who is currently unemployed and has a substantial amount of accumulated debt, should not be given credit. Once the bank has decided to invest in analytics to automate its credit decisions, it can only start from a prefiltered or in other words biased data sample, since this sample will never contain unemployed customers with big accumulated debt. Since the AI model has never seen this type of customers, it will not know that they should be considered as bad payers. This is the so called reject inference problem in credit scoring which essentially boils down to a data bias problem. Several solutions have been developed for this. Some have even suggested to grant credit to everyone during a limited timespan so as to be able to capture the total unbiased population [Thomas et al., 2002]. This is obviously a controversial strategy though its proponents argued that its increased cost is worth the investment of obtaining a better credit risk model. However, the most feasible and commonly used strategy to tackle this reject inference bias is to gather external data (e.g. from credit bureaus) about these customers who were denied credit in the past. In fact, when thinking a bit further about the problem, another form of bias is also present in historical credit data. Banks not only lack information about past rejects, but also about past withdrawals, i.e., customers that didn't take up the offer because they found a better (usually cheaper) one elsewhere. To summarize, the lack of information on historical rejects and withdrawals leaves credit risk modelers with a biased data sample to start developing credit risk models [Baesens et al., 2016].

## **1.5 TYPES OF VARIABLES**

A variable is a data item representing an atomic piece of information that can be used for analytics. Examples are a customer's name, age, income, profession, etc. Different types of variables should be properly distinguished such that summary statistics and analytical techniques are always calculated and estimated in a meaningful way.

A continuous variable can take on any value in a specified interval. This interval can be limited, e.g. between 0 and 1, between 100 and 1000, or can even be unlimited, e.g. between minus infinity and plus infinity. Popular examples are income, savings balance, credit score, age etc. For these variables, it makes sense to calculate summary statistics such as the average, median, standard deviation, and confidence intervals (see Section 1.10).

Categorical variables can only assume values from a predefined set. A further distinction can be made between nominal, ordinal and binary categorical variables. Nominal variables are categorical variables whereby you do not have an ordering between the values. Examples are purpose of loan which can be car, house, cash, SIC industry code, etc. For nominal variables, there is no meaningful ordering between the different values of the variable. Ordinal variables, on the other hand, are variables that are categorical and where there is an ordering between the values. Think about a credit rating, for example. A credit rating can be AAA, which is better than AA, or AA which is in turn better than A, and so on Binary variables are a special case of categorical variables since they can only have two values. An example is employment status, which can either be employed or unemployed. Note that for categorical variables, it does not make any sense to compute statistics such as standard deviation or confidence intervals. It does make more sense to look at statistics such as the mode, which is the most frequently occurring value as we discuss in Section 1.10.

## 1.6 VARIABLE DEFINITION

Each variable needs to be unambiguously, clearly and correctly defined to be used. In most analytical models (e.g., regression models), the output depends in a smooth, continuous way on the continuous input variables. Hence, when using ratios, the ratio definition needs to guarantee that the predicted output can be obtained as a continuous function of the ratio variable. In linear scoring functions, one may encounter difficulties with ratios in which the denominator may change sign. Consider the ratio debt/average net earnings, where the numerator is always positive, but the denominator can be both positive and negative. Negative denominators already indicate stressed companies. For reasonable values of the ratio, the risk increases with higher ratio values. The higher the ratio, the longer it will take to repay the debt. However, when the denominator becomes negative, the ratio becomes negative as well and it loses its interpretation as the (theoretical) number of years needed for debt repayment. More important is that the higher risk now corresponds to a lower ratio value. Such a discontinuous behaviour cannot be captured by most analytical techniques, unless it is explicitly taken into account in the model formulation. One solution in the ratio definition is to replace all negative ratio values by the maximum value or a higher default value. Alternatively, observe that the reversed ratio net earnings/debt does not suffer from the difficulties with discontinuous behaviour. Although it may be less intuitive to apply, it also allows to penalize further for decreasing negative net earnings.

For ratios in which both numerator and denominator can have positive and negative signs, the situation becomes even more complicated. A positive ratio value may be due to both a negative numerator and denominator, or both a positive numerator and denominator. The ratio itself does not allow to distinguish between good and bad value. Such issues can be important in case of automated data feeds that are limited to ratios only. As both numerator and denominator can have positive and negative values, inverting the ratio does not solve the problem. One solution is to put all ratios with a negative numerator or denominator equal to zero or another negative value. Alternatively, one can also restrain the denominator between a small value close to zero and a sufficiently high value. Another solution is to correct the ratio definition where possible. E.g., the scissors ratio income growth to expenses growth with numerators and denominators that can become positive and negative can be replaced by the difference: income growth minus expenses growth.

Apart from the discontinuities in the relation between risk and ratio, also the ratio distribution itself is important. Some ratio definitions tend to result into fat-tailed distributions. This can be explained by the fact that it is well-known that the ratio of two Gaussian distributions is a Cauchy distribution, for which the mean does not exist and with fat tails. When exceptional ratio values occur more frequently than standard values, one should reconsider the ratio definition to make the numerical values more in line with the interpretation for model usage.

## 1.7 TARGET DEFINITION

### 1.7.1 Introduction

Supervised learning or predictive analytics aims at predicting a target label which can either be continuous (e.g., CLV, sales, losses, etc.) or categorical (e.g., fraud, credit default, churn, etc.). This target label will then be used to steer the model (e.g., logistic regression, neural network, XGBoost) learning process. It is obvious that during data preprocessing careful consideration is needed upfront to make sure the target is properly and unambiguously defined. Caution is warranted to make sure the target keeps measuring what it needs to measure. To guarantee the proper alignment between a target label and the accompanying business goal, close collaboration is needed between the AI model developers, data scientists, etc. on the one hand and the business experts on the other hand. This is especially important since any discrepancy between both may be further exacerbated during the model estimation. In fact, most of the target labels we end up defining end up being proxies for what we actually want to know or optimize for [Baesens and

vanden Broucke, 2021].

### 1.7.2 Predictor versus Target Time Measurement

The idea of supervised learning is to use predictors to predict a target both measured at a particular moment in time. Deciding upon how to define both time measurements is not as straightforward as it may seem. As a first approach, let's say we start from the setup depicted in Figure 1.2.

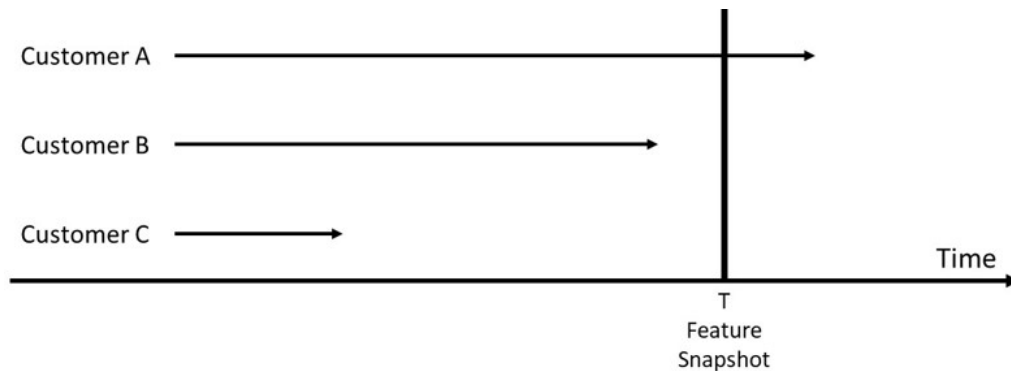


Figure 1.2: Predictor and target snapshot: option 1.

The figure shows three customers: A, B and C. A snapshot of the customers' predictors is taken at time  $T$ . Very often, this is based on the actual, current state as is stored in e.g. an operational data base. The target is then set by inspecting the history of each customer. Assume we are predicting churn or in other words if customers leave the company or not. Customer A is still active at time  $T$  so (s)he would be considered as a non-churner. Customers B and C have churned so they get a positive target. This predictor and target measurement approach is essentially flawed since we are using future information to predict past events. Hence, it would not be unexpected if the resulting churn model performs exceptionally well. More specifically, the R feature of the RFM framework (see Section 1.18.2) measuring customer activity would be a very powerful predictor as for all past churners it would simply state there was no activity. Obviously, a model like this is completely useless and will performance wise break down when put into production.

A better empirical setup is depicted in Figure 1.3. The difference is that the predictors are now measured

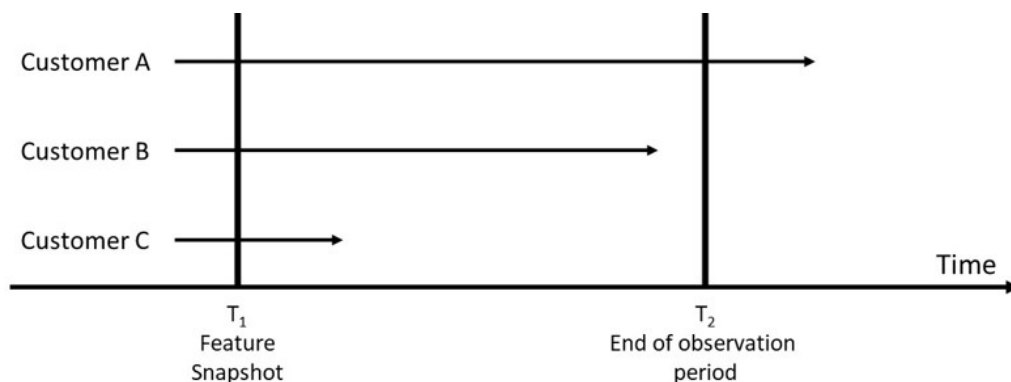


Figure 1.3: Predictor and target snapshot: option 2.

in the past and the target after that. More specifically, an observation period after the predictor measurement is defined during which the target will be determined. The length of this period will obviously determine how many target observations will become available (e.g., how many churners, fraudsters, defaulters, etc.). In Figure 1.3, customers B and C churn during the observation period whereas customer A will churn after that and thus be labelled as a non-churner.

A third option is to include a drop out period as illustrated in Figure 1.4. This is a period during which no targets are measured corresponding to the period between  $T_1$  and  $T_2$  in the figure. The motivation for this is to exclude overly predictive predictors since these may result into high model performance but have been measured too close to the target label in order to be actionable (i.e., prevent churn) to the firm. This implies customer C is left out from the data as the churn event happened too close to the predictor measurement. Consider the example of a prepaid Telco setting. Peak usage of the remaining credit is a very predictive indicator for upcoming churn but typically indicates the customer already made up his mind to leave and decided to churn anyway. Hence, it is essentially too late for the company to act on this and it would be nicer to capture signals of churn such as customer dissatisfaction much earlier so as to be able to act upon it using churn prevention strategies.

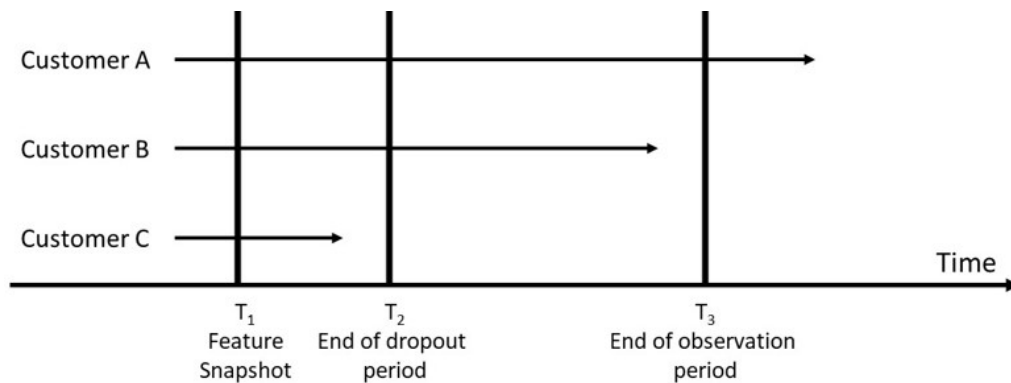


Figure 1.4: Predictor and target snapshot: option 3.

A longer observation period obviously results into more target observations and thus facilitates robust model estimation as it can avoid highly imbalanced data sets. However, this implies that the predictors become more outdated as well which may hamper their predictive performance. Depending upon the business application, the target may be observed for either every customer or a selection of them. For example, in the case of churn prediction every customer churns eventually. However, in credit risk and fraud detection not everyone becomes a defaulter or fraudster, respectively.

#### Target observation periods.

We give some examples of our own industry experience. In our collaborations with Telco firms, we often witnessed a dropout period of one month followed by an observation period of one to three months. For credit risk modeling, we often see no dropout periods and observation periods of 12 to 18 months.

### 1.7.3 Target label definition: challenges

Coming up with a good target definition is not an easy endeavor. We often witnessed a focus on short term goals sometimes even with negative consequences. Let's give some examples.

Churn prediction is often referred to as retention modeling where the aim is to retain your customers as much as possible and deepen their customer relationship if possible [Baesens and De Caigny, 2022]. A churn prediction model essentially tries to predict which customers will leave the company or decrease their product/service usage. A distinction can be made between active churn, passive churn, forced churn and expected churn. Active churn implies that the customer stops the relationship with the firm and switches to another firm. Identifying active churn implies a subscription or contractual setting (e.g., Netflix, postpaid Telco, etc) such that the customer can explicitly cancel the subscription/contract. Passive churn occurs when the customer stays with the firm but decreases the intensity of the relationship (e.g., sleeping bank accounts). Passive churn can be measured both in a subscription/contractual as well as non-subscription/non-contractual setting (e.g., Amazon, Netflix, supermarkets, hotels, gaming. Forced



churn implies that the company stops the relationship with the customer, for example because (s)he has been engaged in fraudulent activities. Forced churn can also occur if a company wants to focus on more profitable customers or avoid having customers with a negative CLV. Expected churn occurs when the customer no longer needs the product or service (e.g., baby products).

The goal of response modeling is to measure the outcome of a marketing campaign. A distinction can be made between an implicit and explicit response. Examples of implicit responses are: reading an advertisement email, clicking on a link, downloading a product description, configuring a product such as a sport shoe or car for example, or contacting the customer service desk for a price quote. Examples of explicit response are a purchase, a purchase and a good product or service review, think about an Amazon book review for example, and, finally, the ultimate response, a purchase, a good review and very active word-of-mouth so as to hopefully create a viral response effect.

In fraud detection, it might sound desirable to label instances (claims, transactions, etc.) as fraudulent based on formally confirmed historical occurrences of fraud [Baesens et al., 2015]. Unfortunately a formal closure and confirmation of fraud can take a long time in many settings and often even never be fully reached. Hence, this leads to the issue of not having enough positive cases to train on, so that the decision is often made to consider suspicious cases as positive instead. Although some of these might never result in a formal confirmation, at least such a model could already be useful. A suspicious case will typically lead to additional efforts and administrative costs being spent in order to e.g. collect evidence, gather documentation, bring it forward to a court case, etc., so that having a model which can output a shortlist would already be beneficial. However, note how the target definition has changed from fraud to suspicion of fraud.

Credit Risk is typically decomposed into three measures: Probability of Default (PD), Loss Given Default (LGD) and Exposure at Default (EAD) (Baesens et al., 2016). The PD measures the probability of the obligor running into default in the upcoming year. The EAD represents the amount owed by the obligor to the lender. The LGD then measures the percentage of EAD likely to be lost upon default. All three credit risk measures come with their own target label challenges. PD modeling is typically tackled as a binary classification problem assuming a preset definition of default which is typically 90 days in payment arrears as set in the Basel accord [Baesens et al., 2016]. LGD modeling starts from a historical data set of defaulted obligors and calculates for each the corresponding losses. However, these losses should take into account indirect costs and benefits (e.g. arrears penalties) as well as proper discounting which pose some real challenges in terms of proper target definition. Similar challenges apply to EAD modeling.

Consider the example of the bounce rate in web analytics. Initially, this was defined as the percentage of site visits where the visitor left instantly after having seen one page. Later on other measurements were introduced such as defining bounces as visits which lasted less than 10 seconds. In fact, this may be a very poor proxy for what we want to measure which is customer satisfaction or engagement. Content websites may want to aim for customers to stay longer on their sites (often referred to as site stickiness) so as to maximize their advertisement generating revenue. Support websites on the other hand want to have customer stay as short as possible as this would imply their problem was quickly and efficiently solved resulting in a positive customer experience.

As another example, in recommender systems as adopted by Netflix or Amazon, we want to provide recommendations that are of interest to customers. This is usually done using user-user collaborative filtering (finding like minded users) or item-item collaborative filtering (finding similarly rated items) (Ricci et al., 2022). However, measuring user interest in this way may miss out on the opportunity of serendipity often defined as the occurrence of events by chance in a happy or beneficial way. In a recommender setting, this translates to recommending a product or service, which the user was not aware of and thus not looking for, but turns out to be very interesting to him/her, in other words, the unexpected, not sought for, yet pleasant surprise.

**Target definition.**

In the past, Google utilized the number of hours users spent watching YouTube as a proxy for how happy they were with the content. Sadly, engineers also found out that this had the unwanted side effect of incentivizing conspiracy theory videos, since convincing users that the mainstream media is lying typically keeps them watching more YouTube as such confirming the recommendation engine’s success.

To summarize, deciding upon a target label definition should be carefully done in collaboration with all stakeholders involved and if possible also even the end customers or subjects being targeted themselves.

**1.8 DENORMALIZING DATA**

Traditional analytical techniques such as regression, decision trees, *k*-nearest neighbor, etc. assume data to be represent in a tabular format representing all the data in a structured way. A structured data table enables straightforward processing and analysis. Typically, the rows of a data table represent the basic entities to which the analysis applies (e.g. customers, transactions, firms, claims, cases, etc.). The rows are also called instances, observations, or lines. The columns in the data table contain information about the basic entities. Plenty of synonyms are used to denote the columns of the data table, such as (explanatory) variables, fields, characteristics, indicators, features, etc. Denormalization refers to the merging of several normalized source data tables into an aggregated, denormalized data table. Merging tables involves selecting information from different tables related to an individual entity, and copying it to the aggregated data table. The individual entity can be recognized and selected in these tables by making use of (primary) database keys, which have been included in the table to allow identifying and relating observations from different source tables pertaining to the same entity. Denormalization is illustrated in Figure 1.5.

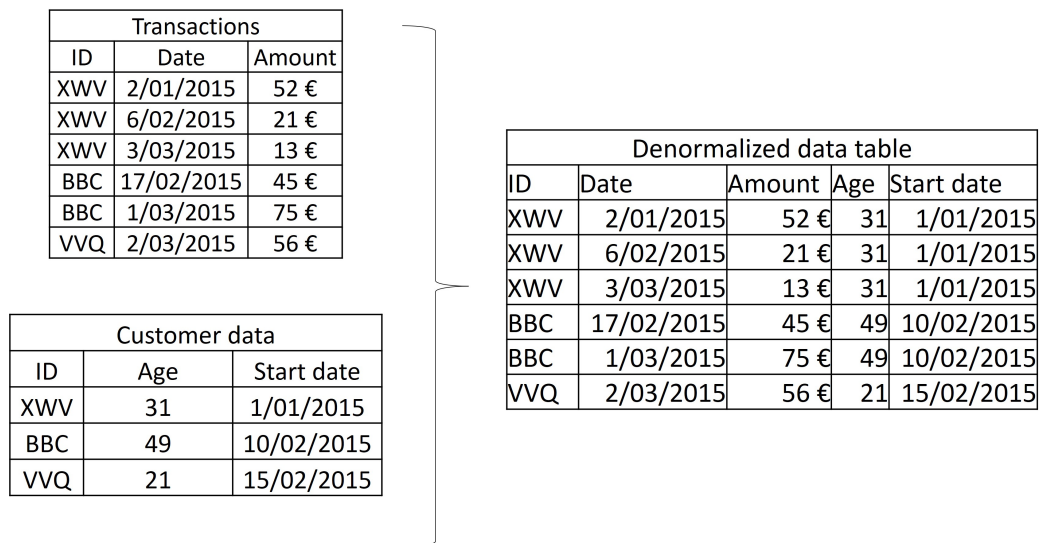


Figure 1.5: Denormalizing Data.

**Normalization and Denormalization.**

Many data sources are stored in normalized relational databases such as MySQL, Oracle, IBM, etc. The idea of normalization is to distribute the data across multiple relational tables connected with primary-foreign key relationships. The benefit of this is that it reduces data inconsistencies and redundancy. For more information, we refer to [Lemahieu et al., 2018].

## 1.9 VISUAL DATA EXPLORATION

Visual data exploration is a very important step of getting to know your data in an ‘informal’ way. It allows gaining initial insights into the data which can be usefully adopted throughout the analytical modeling stage. Different plots/graphs can be useful here, such as bar charts, pie charts, histograms, scatter plots, etc.

A pie chart represents a variable in a circular way, where the whole circle represents the total percent of the data, and the sections the portion of the total percent consumed by each value of the variable. Figure 1.6 shows an example of a pie chart for a residential status variable. Note how most of the customers have their own property.

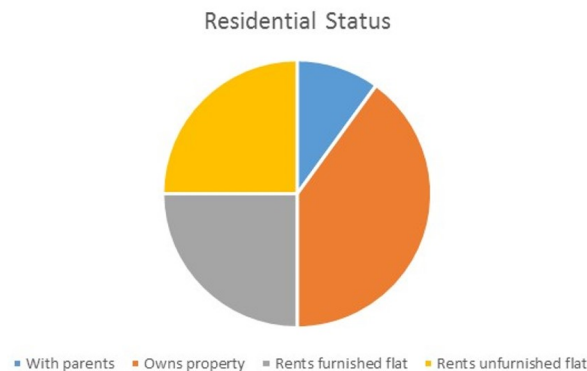


Figure 1.6: Pie chart.

A histogram provides an easy way to visualise the central tendency and to determine the variability or spread of the data. Histograms also provide a means of comparing observed data with certain standard known distributions (e.g. Normal). Figure 1.7 shows an example histogram for the age variable. Note the skew distribution of the age variable and some possibly weird values in the 70+ category.

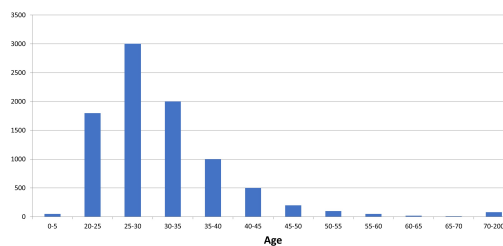


Figure 1.7: Histogram.

Scatter plots allow to visualize one variable against another one to see whether there are any correlation patterns in the data. Figure 1.8 shows a scatter plot between duration and age where a weak linear relationship can be seen. Both 2D as well as 3D scatter plots can be considered.

To get deeper visual insight, you could consider any of the plots discussed previously conditioned on a target variable. In Figure 1.9 you see a pie chart for a housing variable, but now split up for the Goods and Bads separately. You can clearly see that more of the Bads are renting and living for free (for example, with their parents or friends) than the Goods, which typically have their own housing facilities. Insights that result from this preliminary visual analysis can then be discussed with the business expert for validation and if needed can even lead to the creation of features as we discuss in Section 1.18.

To summarize, visual data exploration is an important activity that can be used to determine initial insights about the data. These insights can then be used to steer and validate the remainder of the analytical model development process.

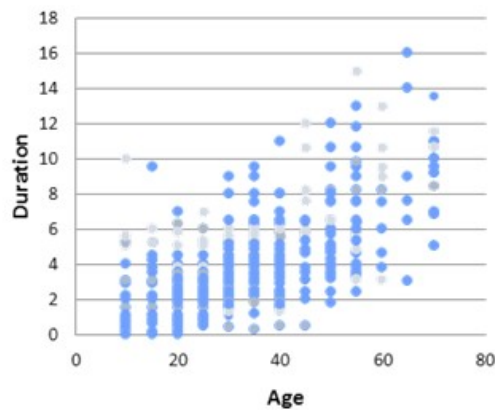


Figure 1.8: Scatter plot.

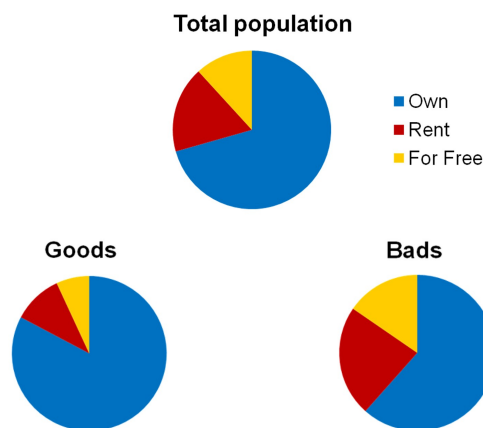


Figure 1.9: Pie chart per class.

### Using Virtual Reality (VR) for Visual Data Exploration.

Virtual Reality (VR) is a promising recent technology for data visualisation. The idea is to use a headset or smart glasses to immerse yourself into your data source(s) and create a 360° degree sphere of space to more naturally interact with your data using as many senses as possible such as vision, touching using haptic feedback gloves allowing you to zoom in/out, touch buttons or move windows, and sometimes even hearing through data-audio relationships as well. The idea of using VR for data exploration has proven especially useful in exploring network or geospatial data. Moreover, it facilitates the collaborative exploration of data by multiple users simultaneously.

## 1.10 DESCRIPTIVE STATISTICS

The next data preprocessing step is to summarize the data by using some descriptive statistics which provide information regarding a particular characteristic of the data. Key tendency descriptive statistics are the mean and median value for continuous variables. The mean is the arithmetic average of all values of a variable. The median is the middle value of a variable, when the values are arranged from low to high. It is less sensitive to extreme values than the mean. The mode is the most frequent or most typical value of a variable. The mode is not necessarily unique, since multiple values can result in the same maximum frequency. Unlike the mean, the mode can also be computed for categorical variables. The  $p$ -percentile of a variable is the value such that  $p\%$  of the observations have a value less than this value. The variance is the mean squared deviation of the observations from the mean. It is a measure of dispersion of a

variable. The standard deviation is the square root of the variance and is expressed in the same units of measurement as the variable being evaluated. Both the variance and standard deviation are affected by extreme observations. The interquartile range (IQR) is defined as the difference between the third and first quartile,  $Q3 - Q1$ . It represents the range of the middle 50% of the data and is unaffected by extreme observations. The standardised value  $z_i$  is calculated by subtracting the variable mean of each observation and dividing by its standard deviation. By definition, the mean of the standardised values equals 0 and the standard deviation 1. Skewness is a measure of symmetry or asymmetry of a distribution. It measures to which extent the data distribution has a more pronounced left tail or right tail. For a left (right) skewed distribution, the skewness coefficient will be negative (positive), since negative (positive) deviations from the mean outweigh the positive (negative) deviations. The mode will then typically be higher (lower) than the median, and mean. For a symmetric distribution, the skewness coefficient will be zero. The kurtosis measures the relative peakedness or flatness of a distribution. Positive kurtosis indicates a peaked, leptokurtic distribution. Negative kurtosis indicates a flat, platykurtic distribution

It is important to note that all these descriptive statistics should be assessed together or in support and completion of each other. For example, comparing the mean and median can give insight into the skewness of the distribution and outliers. Furthermore, the statistics can also be considered for different subpopulations or strata in your data. For example, in fraud detection, they can be considered for the fraudsters and non-fraudsters separately as this may reveal some interesting starting insights into what differentiates them. In fact, they can then be further contrasted statistically using hypothesis or Analysis of Variance (ANOVA) testing procedures. The results from these analyses can then help in defining predictors.

<b>mean</b>	$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
<b>median</b>	Sort data from low to high: $x_{(1)} \leq x_{(2)} \leq x_{(3)} \leq \dots \leq x_{(n)}$ If $n$ is uneven: $x_{Med} = x_{(\frac{n+1}{2})}$ If $n$ is even: $x_{Med} = \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2}$
<b>mode</b>	value with highest frequency
<b>p-percentile</b>	$x_p : \frac{\text{number of } x\text{-values} \leq x_p}{n} \geq p$ $x_{0,5}$ : Median $x_{0,25}$ : First Quartile $x_{0,75}$ : Third Quartile $x_{0,10}, x_{0,20}, x_{0,30}, \dots, x_{0,90}$ : deciles
<b>variance</b>	$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
<b>standard deviation</b>	$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$
<b>Interquartile Range (IQR)</b>	$x_{0,75} - x_{0,25}$
<b>Standardised value</b>	$Z_i = \frac{x_i - \bar{x}}{s}$
<b>Skewness</b>	$\frac{1}{n-1} \sum_{i=1}^n z_i^3$
<b>Kurtosis</b>	$\frac{1}{n-1} \sum_{i=1}^n z_i^4 - 4$

Table 1.1: Key descriptive statistics.

## 1.11 MISSING VALUES

Missing values can occur for various reasons. The information can be non-applicable. For example, when modeling the date of default, this information is only available for the defaulters and not for the non-defaulters since it is not applicable there. The information can also be undisclosed, such as a customer who has decided not to disclose his or her income because of privacy. Missing data can also originate from an error during merging (e.g., typos in name or ID). In Table ??, you can see a data set with missing values represented by question marks. Some analytical techniques cannot work with missing values and

ID	Age	Income	Marital Status	Credit Bureau Score	Class
1	34	1800	?	620	Bad
2	28	1200	Single	?	Good
3	22	1000	Single	?	Good
4	60	2200	Widowed	700	Bad
5	58	2000	Married	?	Good
6	44	?	?	?	Good
7	22	1200	Single	?	Good
8	26	1500	Married	350	Good
9	34	?	Single	?	Bad
10	50	2100	Divorced	?	Good

Table 1.2: Data set with missing values.

preprocessing them is needed. Others such as decision trees can easily incorporate them during the tree construction process.

Three common strategies to deal with missing values are: Keep, Delete and Replace. The Keep strategy is recommended in case missing values are meaningful because they indicate a meaningful pattern. As an example, a missing value for income could imply unemployment, which may be related to credit risk default. For categorical variables, we may include an additional category for the missing value. Another option would be to add an additional missing value indicator variable, either one per variable or one per entire observation. Another option is to delete a variable with too many missing values (e.g., when only less than 20% are known values) or an observation (casewise deletion) with too many missing values. The replace or impute option estimates missing values using imputation procedures. For continuous variables, missing values can be replaced by the mean. Note however that the mean is sensitive to outliers. Hence, a wiser option could be to use the median instead. If missing values can only occur during model development, you can also replace the missing value by the mean or median of all observations within the given target class. For categorical variables, missing values can be replaced by the mode or most frequent category. Again, if missing values only occur during model development, we can replace with the mode of all observations of same class. Note that is important to be consistent when treating missing values during model development and during model usage to make sure that the analytical models learned on them generalize properly.

Regression or tree based imputation procedures are another option to deal with missing values. Here an regression or decision tree (or any other analytical model) is built to estimate the missing values based upon all the other information available. For example, you could build a linear regression or decision tree predicting income based upon age, marital status and credit bureau score. This model can then be used to find the values for the missing income observations. Although this technique seems powerful, empirical evidence has shown that it often does not substantially add to the performance of the resulting model. Hence, it is perfectly fine to continue with the simpler mean/median/mode imputation approaches we discussed earlier.

## 1.12 OUTLIERS

Outliers are extreme or unusual observations that are very dissimilar to the rest of the population. Two types of outliers should be considered: valid observations (e.g., the CEO's salary is 1,000,000 US \$) and invalid observations (e.g., age is 300 years). Some techniques, like decision trees, are robust with respect to outliers and require no additional preprocessing. Others, such as linear and logistic regression, are more sensitive to them. Two important steps in dealing with outliers are detection and treatment.

Simply calculating the minimum or maximum of a variable can be a first simple yet effective step in verifying the presence of outliers. A histogram as visualised in Figure 1.10 can give a complete picture.

Notice the outlying observations at both extremes of the plot (age categories [0,5] and [150,200]).

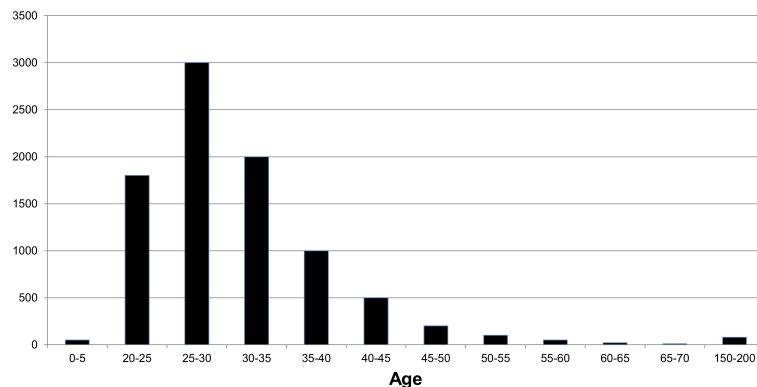


Figure 1.10: Age distribution: outliers.

Another useful visual mechanism are box plots as illustrated in Figure 1.11. A box plot represents three key quartiles of the data: the first quartile (25% of the observations have a lower value), the median (50% of the observations have a lower value) and the third quartile (75% of the observations have a lower value). All three quartiles are represented as a box containing 50 % of the data. The minimum and maximum value are then also added unless they are too far away from the edges of the box. Too far away is then quantified as more than 1,5 times Interquartile Range ( $IQR = Q3 - Q1$ ). In the box plot three outliers can be seen.

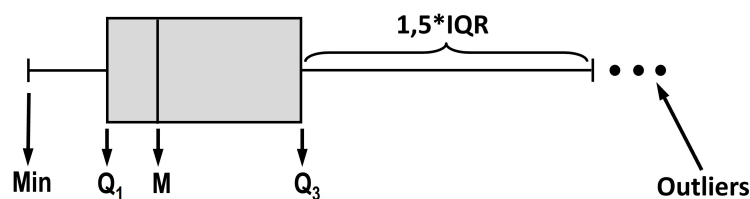


Figure 1.11: Box plot.

Another way is to calculate the z-scores, measuring how many standard deviations an observation lies away from the mean by calculating the standardized values  $z_i$  as shown in Table 1.1. Note that by definition the  $z_i$ -scores will have 0 mean and unit standard deviation. A practical rule of thumb then defines outliers when the absolute value of the z-score  $|z|$  is bigger or equal than 3. Since z-scores both use the mean and standard deviation, they are sensitive to outliers. A robust alternative can be used based on the median and interquartile range (IQR) as follows:

$$z_i^{rob} = \frac{x_i - Median}{IQR} \quad (1.1)$$

Some analytical techniques (e.g. decision trees, neural networks, SVMs) are fairly robust with respect to outliers due to built-in mechanisms such as weight regularisation. Others (e.g. linear/logistic regression) are more sensitive to them. Various schemes exist to deal with outliers. It highly depends upon whether the outlier represents a valid or invalid observation. For invalid observations (e.g. age is 300 years), one could treat the outlier as a missing value using any of the schemes discussed in the previous section: keep, delete, replace. For valid observations (e.g. income is \$1.000.000), other schemes are needed. A popular scheme is truncation also called capping or winsorizing. One hereby imposes both a lower and upper limit on a variable and any values below/above are brought back to these limits. The limits can be calculated using the z-scores. An example would be to replace all values with z-scores  $\geq 3$  with the mean plus 3 times the standard deviation, and all values with z-scores less than -3 with the mean minus three times the

standard deviation. Also expert based limits based upon business knowledge and/or experience can be imposed. In Figure 1.12 you can see capping based on the z-scores illustrated.

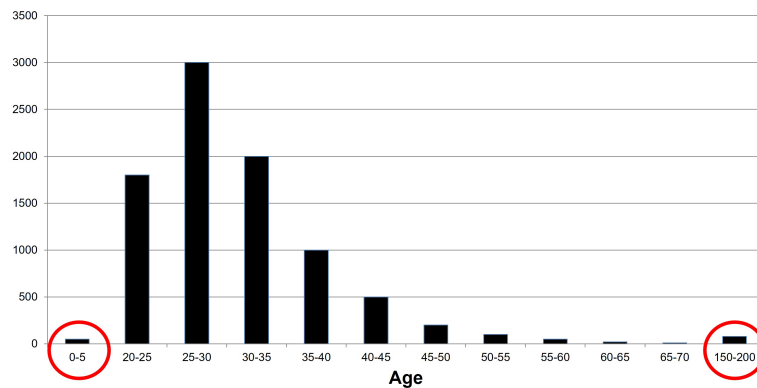


Figure 1.12: Capping based on z-scores.

## 1.13 STANDARDIZING DATA

Standardization (also known as normalization) involves transforming the values of a variable to another range. It may be needed because variables are typically expressed on different scales. As an example, think about income measured between Euro 1,000 and Euro 1,000,000 and debt ratio measured between 0.1 and 1. Some analytical methods are scale dependent. A popular example is clustering which is based on calculating (e.g., Euclidean) distances. In order to avoid that one variable outweighs another because of the scaling, standardization can be performed. Furthermore, standardization might also be necessary in order to avoid numerical problems due to arithmetic overflows, or to speed up the parameter estimation process. It can also help to interpret and compare the predictive impact of variables.

Different standardization methods might be used. Min-max standardization linearly standardizes a variable  $x$  to the range  $[newmin, newmax]$ :

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}(newmax - newmin) + newmin \quad (1.2)$$

Common values for  $newmin$  and  $newmax$  are 0 and 1, or -1 and 1; respectively. Another way to do standardization is by using the z-scores discussed in Table 1.1. This method is sometimes referred to as zero-mean standardization and will work well when one does not actually know the minimum or maximum of the variable. Decimal scaling standardization transforms a variable to the range  $[-1; 1]$  and works by moving the decimal point as follows:

$$x' = \frac{x}{10^k} \quad (1.3)$$

where  $k$  is the smallest integer such that  $\max(|x'|) < 1$ .

## 1.14 CATEGORICAL VARIABLE CODING

Special care should be given to how categorical variables are represented in predictive models. In what follows, we discuss different ways of coding categorical variables.

Metric coding is a first option to code categorical variables by assigning numeric values 1, 2, 3, ... to each of the categories. For nominal variables, this way of coding imposes an unwanted ordering on the values which is clearly not appropriate. For ordinal variables, like, e.g., credit ratings, this implies that the



Purpose of loan	D1	D2	D3	D4	D5
Car	1	0	0	0	0
Cash	0	1	0	0	0
Travel	0	0	1	0	0
Study	0	0	0	1	0
Business	0	0	0	0	1

Table 1.3: One hot encoding a categorical variable.

difference between any two adjacent ratings is the same, no matter where the ratings are situated on the rating scale. The use of metric coding with linear models is advised against.

Dummy coding is more common and meaningful way of coding categorical nominal variables with no scale assumption present. Table 1.3 provides an example of how the purpose variable with 5 values is represented using 1-of- $n$  dummy coding also called one-hot encoding.

Because the last dummy variable can be derived based on the values of the previous dummies and the intercept, a popular coding scheme is 1 – of –  $n – 1$  coding with  $n-1$  dummy variables, where one value is defined as the reference category. This reference category does not get any explicit weight by the model. The weights of the other categories are then measured in an incremental way vis-à-vis the reference category as shown here:

$$y = \beta_0 + \beta_1 D_1 + \beta_2 D_2 + \beta_3 D_3 + \beta_4 D_4 \quad (1.4)$$

The category Business is the reference category and gets a weight of  $\beta_0$  whereas the category Car gets a weight of  $\beta_0 + \beta_1$ . Note that also other encoding schemas exists such as effect or deviation coding, contrast coding, etc., but these are less popular.

## 1.15 CATEGORIZATION

### 1.15.1 Introduction

Categorization is also known as coarse-classification, classing, grouping, and binning. The aim is to group values of variables into categories [Baesens et al., 2016]. It can be meaningful for both categorical as well as continuous variables.

The goal of using categorization for categorical variables is to reduce to number of categories to a manageable size so as to avoid having to use too many 0/1 or other dummy indicators when using 1 – of –  $n – 1$  coding. This will make the estimates more robust since we now need to estimate less parameters with the same amount of data. In other words, categorization for categorical variables aims at grouping the values of a categorical variable into a few categories in order to get a more concise and thus more robust model.

Categorization of continuous variables can be beneficial to introduce non-linear or non-monotonic effects into linear models such as linear and logistic regression. Consider the default risk versus age relationship depicted in Figure 1.13.

This pattern was found by our colleague and friend prof. Lyn Thomas from the University of Southampton [Thomas et al., 2002]. As you can see, the graph is non-monotonically decreasing. It decreases until around the age of 26, then increases back again until around 32, followed by a more or less monotonic decrease. Many reasons can be thought of for the risk increase starting at the age of 26. Examples are kids, marriage or may be even an early mid life crisis. Now, if we would approach this non-linear pattern with a linear regression model, the best fit would be a linear decreasing line. This would imply that we miss out on the local non-monotonic behavior in the age range between 26 and 32. By categorizing the age variable into 3 categories: less than 26, 26 to 32 and 32 plus, we can have separate regression coefficients for each of

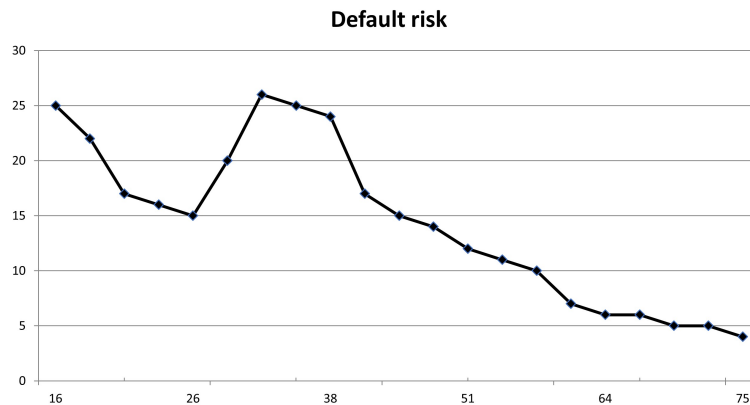


Figure 1.13: Default risk versus age.

these categories and thus approximate the non-linear behavior in a piece-wise linear way. A disadvantage however is the loss of information, although this represents information that cannot be successfully used by the linear regression model anyway. Note however, that if you were using non-linear models such as neural networks, then categorization would not be needed as these models can directly cope with the non-linear patterns in the data.

### 1.15.2 Categorization methods

In what follows, we discuss five popular methods for categorization: binning, pivot tables, Chi-squared analytics, business knowledge and decision trees.

A first very basic approach towards categorization is binning. Let's assume we have an income variable which has only 6 values: 1000, 1200, 1300, 2000, 1800, and 1400. Equal interval binning creates bins by splitting the variable's range into equal sized parts. Since the range is from 1000 to 2000 and assuming we want two bins, the first bin would then be from 1000 to 1500 and the second one from 1500 to 2000. Note that the first bin has 4 values and the second bin has 2 values. Equal frequency binning creates bins based upon the frequency. In other words, it ensures that every bin has the same number of values. In our case, assuming we create 2 bins, the first bin contains the lowest 3 values and the second bin the highest 3 values. Equal frequency binning will be more useful if outliers have not yet been removed since these may have too big an influence in equal interval binning. It is clear that both these binning approaches are very basic in the sense that they do not take into account any target variable such as credit risk default or fraud during the binning process. They just look at variables individually, such as income in our case.

A second technique to perform categorization is pivot tables as Figure 1.14 illustrates.


Here you can see an example of a credit scoring data set with a categorical purpose variable. A pivot table can now be created to compute the number of goods and bads for each of the purpose values. For example, for car, the number of goods equals 1000, the number of bads equals 500. This will then in turn allow to calculate the odds, which is simply the number of goods divided by the number of bads resulting in an odds of 2. We can now perform categorization by grouping purpose values with similar odds. Let's say we want 3 groups for our example. Group 1 can then consist of car and study, since both have the lowest odds, group 2 is house, and group 3 is cash and travel since both have the highest odds.

A more sophisticated method for doing categorization is based on Chi square analysis. Let's illustrate how this works [Thomas et al., 2002]. In Table 1.4 you can see a variable residential status, which has values owner, rent unfurnished, rent furnished, with parents, other and no answer. The table depicts the distribution of goods and bads across these values.

Let's say we are considering two options for categorization: option 1 is owners, renters and others

Customer ID	Age	Purpose	...	Bad/Good
C1	44	car	...	Good
C2	20	cash	...	Good
C3	58	travel	...	Bad
C4	26	car	...	Good
C5	30	study	...	Bad
...	...	...	...	...

Pivot Table



	Car	Cash	Travel	Study	House	...
Good	1000	2000	3000	100	5000	...
Bad	500	100	200	80	800	...
Odds	2	20	15	1,25	6,25	...

Figure 1.14: Pivot tables for categorization.

Attribute	Owner	Rent		With parents	Other	No answer	Total
		Unfurnished	Furnished				
<b>Goods</b>	6000	1600	350	950	90	10	9000
<b>Bads</b>	300	400	140	100	50	10	1000
<b>Goods/ Bads odds</b>	20:1	4:1	2.5:1	9.5:1	1.8:1	1:1	9:1

Table 1.4: Residential status variable.

whereas option 2 is owners, with parents and others. We will now analyze which is the best categorization option using Chi-square analysis. Let's first zoom into option 1. We start by building a table with the empirical frequencies for option 1 (see Table 1.5). The number of good and bad owners can be directly copied from the previous table. The number of good renters is the sum of the number of good unfurnished renters, which was 1600, and the number of good furnished renters, which was 350, thus equaling 1950 in total. The other numbers in this table are computed in a similar way.

We now contrast these empirical frequencies with the independence frequencies as shown in Table 5. The independence frequencies are the frequencies that are obtained by assuming that the good/bad status is independent from the residential status. In other words, assuming independence between both variables, the number of good owners becomes  $6300/10000 \times 9000/10000 \times 10000$  or 5670 as you see depicted in the table below.

Once we have all independence frequencies calculated, we can compare them with the empirical frequencies. Assume we would have a perfect match of the numbers in both tables. This would imply that what we observe empirically is independence, or, the good/bad variable is independent from the residential status variable. This is clearly not what we want since we want both to be dependent as much as possible to have a good categorization. In other words, we want to have the empirical frequencies as different as possible from the independence frequencies. The Chi square statistic is a measure which quantifies the

Attribute	Owner	Renters	Others	Total
<b>Goods</b>	6000	1950	1050	9000
<b>Bads</b>	300	540	160	1000
<b>Total</b>	6300	2490	1210	10000

Table 1.5: Empirical frequencies for option 1 for categorizing the residential status variable.

Attribute	Owner	Renters	Others	Total
Goods	5670	2241	1089	9000
Bads	630	249	121	1000
Total	6300	2490	1210	10000

Table 1.6: Independence frequencies for option 1 for categorizing the residential status variable.

difference between the empirical and independence frequencies. It is calculated by summing the ratio of the squared difference between the empirical and independence frequencies and the independence frequencies across all cells of the table

$$\chi^2 = \frac{(6000 - 5670)^2}{5670} + \frac{(300 - 630)^2}{630} + \frac{(1950 - 2241)^2}{2241} + \frac{(540 - 249)^2}{249} + \frac{(1050 - 1089)^2}{1089} + \frac{(160 - 121)^2}{121} = 583 \quad (1.5)$$

The bigger this value the bigger the dissimilarity between both tables and thus the more dependence there is between the good/bad variable and the residential status variable. In order to formally judge upon its significance, the obtained Chi square statistic should follow a  $\chi^2$ -square distribution with  $k - 1$  degrees of freedom, with  $k$  the number of classes of the characteristic which is 3 in our case. This can then be summarized by a  $p$ -value to see whether there is a statistically significant dependence or not. We can now also compute the Chi-square value for option 2. Following a similar computation, we can see that the value is 662. Since both options assume 3 categories, we can directly compare the value of 662 to 583 and since the former is bigger conclude that option 2 is the better categorization.

Two other popular methods to do categorization are based on business knowledge or decision trees. An example based on business knowledge concerns the well-known NACE European industry standard classification system and its SIC American counterpart. Both classify economic activities of companies in a hierarchical way and can as such be used for categorization purposes. We discuss decision trees in Chapter ??.

## 1.16 WEIGHTS OF EVIDENCE AND INFORMATION VALUE

Although categorization reduces the number of parameters or weights for the categorical variables, it introduces new additional parameters for the continuous variables. You can see this illustrated in the regression model here with the age and purpose variables.

$$Y = \beta_0 + \beta_1 \text{Age}_1 + \beta_2 \text{Age}_2 + \beta_3 \text{Age}_3 + \beta_4 \text{Purp}_1 + \beta_5 \text{Purp}_2 + \beta_6 \text{Purp}_3 + \beta_7 \text{Purp}_4 \quad (1.6)$$

Note that we still need a lot of dummy variables. Assume now that we want to make the model more parsimonious by estimating only one parameter for age and purpose, respectively. In other words, is there a transformation  $f()$  which we could define which is monotonically related to the target variable  $Y$ , representing the good/bad status.

$$Y = \beta_0 + \beta_1 f(\text{Age}_1, \text{Age}_2, \text{Age}_3) + \beta_2 f(\text{Purp}_1, \text{Purp}_2, \text{Purp}_3, \text{Purp}_4) \quad (1.7)$$

This transformation can either be monotonically increasing or decreasing resulting in either a positive or negative value of the model parameters ( $\beta_0, \beta_1, \dots$ ), respectively. A popular transformation to do this is the weights of evidence (WOE) transformation defined as follows [Baesens et al., 2016, Thomas et al., 2002]:

$$\text{Weight Of Evidence}_{\text{category}} = \ln \left( \frac{p_{\text{good}}_{\text{category}}}{p_{\text{bad}}_{\text{category}}} \right) \quad (1.8)$$

with

$$\begin{aligned} p_{\text{good}}_{\text{category}} &= \frac{\#\text{goods}_{\text{category}}}{\#\text{goods}_{\text{total}}} \\ p_{\text{bad}}_{\text{category}} &= \frac{\#\text{bads}_{\text{category}}}{\#\text{bads}_{\text{total}}} \end{aligned} \quad (1.9)$$

Age	Count	Distr. Count	Goods	Distr Goods	Bads	Distr Bads	WOE
Missing	50	2,50%	42	2,33%	8	4,12%	-57,28%
18-22	200	10,00%	152	8,42%	48	24,74%	-107,83%
23-26	300	15,00%	246	13,62%	54	27,84%	-71,47%
27-29	450	22,50%	405	22,43%	45	23,20%	-3,38%
30-35	500	25,00%	475	26,30%	25	12,89%	71,34%
35-44	350	17,50%	339	18,77%	11	5,67%	119,71%
44+	150	7,50%	147	8,14%	3	1,55%	166,08%
	2000		1806		194		

Table 1.7: Calculating Weights of Evidence (WOE).

Note that if the proportion of goods within a category is bigger than the proportion of bads, then the ratio will be bigger than 1 and the WOE will be positive. Likewise, if the proportion of goods within a category is smaller than the proportion of bads, then the ratio will be less than 1 and the WOE will be negative.

Consider the categorized age variable as depicted in Table 1.7. Assume we have a data set of 2000 observations: 50 customers have a missing value for age, 200 customers have an age between 18 and 22 and so on. 50 out of 2000 equals 2.50% such that 2.50% of the customers have a missing value for age. Let's now look at the goods and bads separately. There are a total number of 1806 goods, 42 of which have a missing value for age equaling 2.33%. There are a total number of 194 bads, 8 of which have a missing value for age equaling 4.12%. The Weights of Evidence, which is the final column, can then be calculated as the logarithm of the distribution of goods divided by the distribution of bads. The result can then be multiplied by 100 to represent it as a percentage. Remember that a logarithmic curve crosses the X-axis at the value of 1, since the logarithm of 1 is zero. So, if the WOE is bigger than 0, it means that the input to the logarithmic transformation is bigger than 1, or that there are more goods than bads within the category. Indeed, you can see this for the age categories 30 to 35, 35 to 44 and 44+. Vice versa, if the weights of evidence is negative.

Based upon the weights of evidence, we can now define a new measure called the information value (IV) which will allow us to determine the predictive power of a variable.

$$IV = \sum_{i=1}^k (p\_good_i - p\_bad_i) \times woe_i \quad (1.10)$$

Let's re-consider Table 1.7. If age would be a predictive variable, you would expect certain age categories where you have a concentration of good payers and other age categories where you have a concentration of bad payers. This very basic intuition can now be quantified by means of the information value. The information value is the sum across all categories of the product of the difference between the distribution of goods and bads, multiplied by the weights of evidence. If for a particular category, the distribution of goods is bigger than the distribution of bads, then the weights of evidence will also be positive and the product will thus be positive. Vice versa, if for a particular category, the distribution of goods is smaller than the distribution of bads, then the weights of evidence will be negative but the product of both will remain positive. In other words, the product tells us something about the absolute difference between the distribution of goods and bads in a particular category. Summed across all categories, it gives us an aggregate measure for the difference. Hence, the higher the value of the information value, the more predictive the variable is, since it gives us more information about the target. The information value measure can now be used in various ways. It can first be used to assess the appropriateness of the categorisation. In fact, it can help to adjust the categorization so as to increase the information value of the categorized variable. It can also be used for variable selection since a higher information value represents a more predictive variable.

For a given number of categories, the category boundaries can be optimized so as to maximize the

predictive power in terms of IV. Suppose we have categorized the age variable in 5 categories, which implies 4 boundaries as depicted in Figure 1.15.

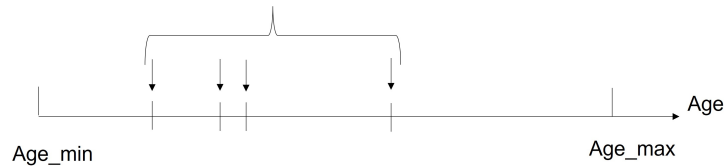


Figure 1.15: Category optimisation using IV.

The data scientist can then shift these 4 boundaries and gauge the impact on the information value of the age variable. Deciding upon the optimal number of categories involves a trade-off. Fewer categories can be preferred, because of simplicity, interpretability, stability and/or robustness. Note however that a single category would imply the loss of all predictive power. On the other hand, more categories allow you to keep as much as possible the predictive power of the variable. For an increasing number of categories, a gain in predictive power is achieved up until some point, when further increasing the number of categories does not add significant predictive power. A practical solution is to perform a sensitivity analysis and simultaneously evaluate both the information value and number of categories. The cut-off can then be set at the elbow point. Also note the fewer values in a category, the less reliable/robust/stable the WOE value that is calculated.

After categorization and WOE coding, 2 modeling strategies can be adopted. First the WOE values can be used as such in the regression model. This would imply that every category has a fixed impact on the default risk as shown in the Figure 1.16. A second strategy would be to include an interaction effect between the WOE and age variable. That would allow us to piecewise approximate the non-linearity as shown Figure 1.17 . Obviously, this would also imply a loss of interpretability.

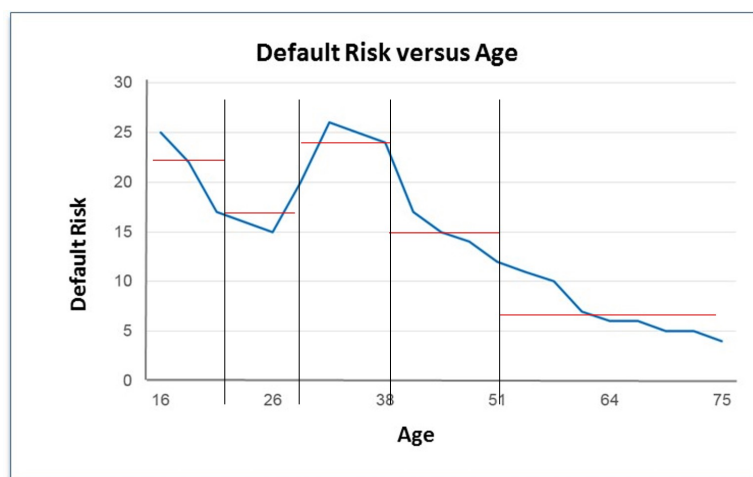


Figure 1.16: Using WOE variables as model inputs: option 1.

## 1.17 DATA QUALITY

To conclude this section on data preprocessing we would like to mention a few things about data quality. We first define it and then elaborate on its dimensions.

### 1.17.1 Definition

Data quality (DQ) is often defined as “fitness for use”, which implies the relative nature of the concept [Moges et al., 2013]. Data that is of acceptable quality in one analytical context may be perceived to be of

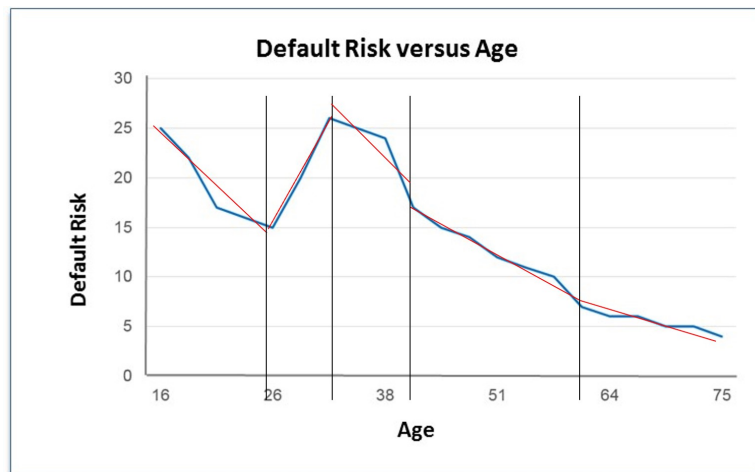


Figure 1.17: Using WOE variables as model inputs: option 2.

poor quality in another context, even by the same data scientist or business user. For instance, the extent to which data is required to be complete for accounting tasks may not be required for sales prediction tasks where approximate estimates may be good enough.

Data quality determines the intrinsic value of the data to the business. Technology such as analytics or AI only serves as a magnifier for this intrinsic value. Hence, high quality data combined with effective analytics is a great asset, but poor quality data combined with effective analytics is an equally great liability. We already referred to this earlier as the GIGO principle, or Garbage In, Garbage Out principle, stating that bad data results into bad models and thus decisions, even with the best technology available. Decisions made based on useless data have cost companies billions of dollars.

#### Impact of Data Quality.

It is estimated that approximately ten percent of customers change their address on a yearly basis (See <https://fivethirtyeight.com/features/how-many-times-the-average-person-moves/>). Obsolete customer addresses can have substantial consequences for mail order companies, package delivery providers or government services since these are typically used in their models. Likewise, it is estimated that on average one in three people change their e-mail address once a year, typically due to switching ISPs, changing jobs or just dodge spammers.

### 1.17.2 Data Quality Dimensions

Data quality is a multi-dimensional concept in which each dimension represents a single aspect or construct, comprising both objective and subjective perspectives. Some aspects are absolute, whereas others depend on the type of task and/or experience of the data user. Therefore, it is useful to define data quality in terms of its dimensions. A data quality framework categorizes the different dimensions of data quality. Different DQ frameworks exist, but a prevalent one is the framework by Wang et al. [Wang and Strong, 1996]. It is represented in Table 1.8 below and shows the different data quality dimensions grouped into four categories.

The framework provides a means to measure, analyze and improve data quality. The intrinsic category represents the extent to which data values are in conformance with the actual or true values. The contextual category measures the extent to which data is appropriate to the task of the data consumer. Obviously, this can vary in time and across data consumers. The representation category indicates the extent to which data is presented in a consistent and interpretable way. Hence, it relates to the format and meaning of data. Finally, the access category represents the extent to which data is available and obtainable in a secure manner. This is especially important in today's networked environment with data being distributed across various platforms. Each category has multiple dimensions as you can see illustrated in the table. High-

quality data should be intrinsically good, contextually appropriate for the analytical task, clearly represented, and accessible to the data consumer.

Finally, a word of caution with regards to data quality: although the GIGO principle certainly applies in practice, it has far too often been used as a popular excuse to explain failed analytics projects. Since all organizations suffer from data quality issues, and improving data quality is not at the top of the list of priorities for most, this is indeed an easy excuse for practitioners. For senior decision makers facing this statement, it is important to try to uncover the true root cause. This might be data quality issues in which case it is important to know which sources of data contained the most problems. However, the underlying problem might also be due to a weak specification of the business problem, lack of solid project management, failure to plan or failing to realize the time required to deliver a good analytical model. It might also be the case that there is simply not enough predictive power present in the data to correlate with the target. One way to deal with this is proper feature engineering as we discuss in the next section.

#### **Impact of Data Quality.**

Spain's newest generation of S-80 submarines were planned to enter service in 2015. However, a weight miscalculation error due to an engineer putting a decimal in the wrong place implied they were 75-100 tons heavier than anticipated, putting them at risk of not being able to resurface after submerging. A complete redesign was needed and the project went over budget more than 30% or about 2 billion Euro approximately and several years of delay in delivery. It clearly shows the importance and impact of data accuracy and representation.

## **1.18 FEATURE ENGINEERING**

### **1.18.1 Definition**

The aim of feature engineering is to transform data set variables into features so as to help AI models achieve better performance in terms of either predictive performance, interpretability or both [Verdonck et al., 2021, Baesens et al., 2021]. Hence, when doing feature engineering it is important to take the bias of your AI technique into account. As an example, a logistic regression assumes a linear decision boundary to separate both classes. Hence, when defining smart features for logistic regression, your aim is to make sure that these new features make the data linearly separable. That will allow the logistic regression to come up with the best model possible. A very simple of feature engineering is deriving the age from the date of birth variable. Feature engineering can be done manually, by the data scientist typically in collaboration with the business user, or fully automated using sophisticated techniques as we discuss later. The importance of feature engineering can not be underestimated. It is my firm conviction that the best way to improve the performance of an analytical model is by designing smart features, rather than focusing too much on the choice of the analytical technique.

To illustrate the potential of feature engineering, let's consider the following example taken from the Kdnuggets website. Suppose you have a supermarket located at the centre of the plot and the most profitable customers live in close proximity to it. It is quite obvious to us humans that we need to consider customers living within a specific radius from the supermarket. This requires knowledge of both the x and y coordinates. For analytical techniques, this data set would be quite difficult to model. For example, logistic regression assumes a linear decision boundary and would not be capable of modeling this data set. Decision tree algorithms would tackle it one variable at a time by creating splits that represent decision lines perpendicular to the axes. To divide the space this way would require a lot of splits.

However, we can perform a simple transformation of coordinates which we were taught in high school. The transformation is from the so-called cartesian coordinates system to the polar coordinates system. This creates the data set as depicted with two smart features, r and theta. You can see that by creating these features, the data set becomes a lot easier to model using any traditional analytical technique such



Category	DQ Dimension	Definition
Intrinsic	Accuracy	Extent to which data is certified, error-free, correct, flawless and reliable.
	Objectivity	Extent to which data is unbiased, unprejudiced, based on facts and impartial.
	Reputation	Extent to which data is highly regarded in terms of its sources or content.
Contextual	Completeness	Extent to which data is not missing, covers the needs of and is of sufficient breadth and depth for the task.
	Appropriate-amount	Extent to which the volume of data is appropriate for the task at hand.
	Value-added	Extent to which data is beneficial and provides advantages from its use.
	Relevance	Extent to which data is applicable and helpful for the task at hand.
	Timeliness	Extent to which data is sufficiently up-to-date for the task at hand.
	Actionable	Extent to which data is ready for use.
	Interpretable	Extent to which data is in appropriate languages, symbols and the definitions are clear.
Representation	Easily-understandable	Extent to which data is easily comprehended
	Consistency	Extent to which data is continuously presented in the same format.
	Concisely-represented	Extent to which data is compactly represented, well-presented, well-organized, and well-formatted.
	Alignment	Extent to which data is reconcilable (compatible).
Access	Accessibility	Extent to which data is available, or easily and swiftly retrievable.
	Security	Extent to which access to data is restricted appropriately to maintain its security.
	Traceability	Extent to which data is traceable to the source.

Table 1.8: Data Quality Dimensions

as logistic regression or decision trees. Obviously, this is a trivial example and with real data, it is rarely that simple, but this shows the potential of proper feature engineering.

### 1.18.2 RFM features

RFM stands for Recency, Frequency, and Monetary and has been popularized by Cullinan already in 1977 in a marketing setting. The RFM features are summarized from transactional data as follows:

- **Recency**: recency of a transaction;
- **Frequency**: frequency of transactions in a given time span;
- **Monetary**: the monetary value of a transaction.

The RFM variables have been very popular in marketing analytics such as customer segmentation, churn prediction and response modeling. You can see an example in Figure 1.18 where a customer transaction database is segmented based upon the RFM variables using quintiles (Recency score 1 = the 20% most recent buyers, Frequency score 1 = the 20% most frequent buyers, Monetary score 1 = the 20% top spenders, etc).

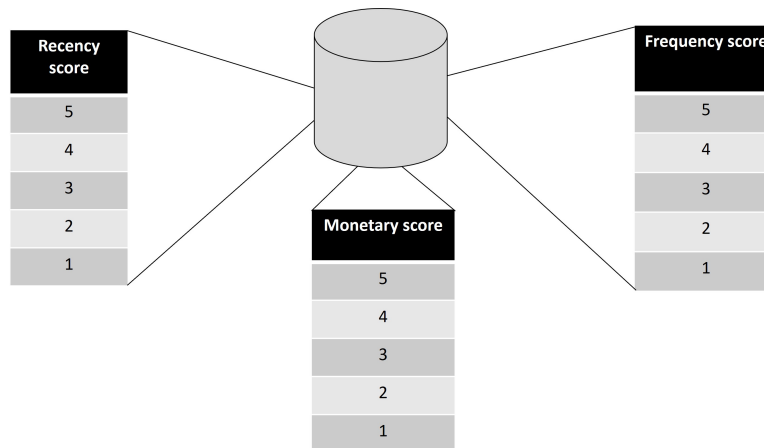


Figure 1.18: Segmenting a customer database using RFM.

Note that in fact each of the RFM constructs can be operationalized in various ways. For example,

- **Recency**: how long ago since a purchase was made? Did a purchase take place the last day/week/month/year, ...
- **Frequency**: how many purchases the last day/week/month/year? What was the minimum/maximum/average/most recent daily/weekly/monthly/yearly frequency?
- **Monetary**: what was the most recent purchase amount? What was the minimum/maximum/average/most recent daily/weekly/monthly/yearly purchase amount?

Besides marketing, the RFM features have also been very popular in fraud detection [Baesens et al., 2015]. Consider the example shown in Figure 1.19 which depicts the usage of authentication methods such as passwords, itsme, fingerprints, iris scans and hardware tokens for payment transactions. The figure shows the time at which Alice made a transfer and the corresponding authentication method she used.

When the time-period between two consecutive transfers with the same authentication method is large, we say that the authentication method has not recently been used. In that case we set recency close to zero as you can see illustrated for AU03. When the time-period between two consecutive transfers with the same authentication method is small, we say that the authentication method has recently been used. In that case we set recency close to one, as you can see for AU01. When an authentication method is used for the

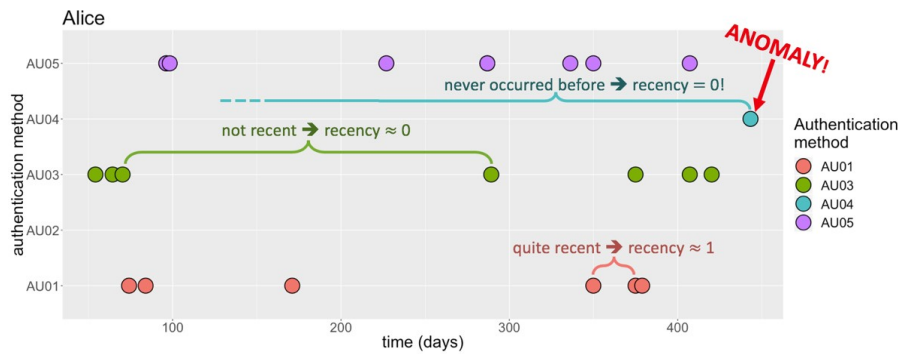


Figure 1.19: RFM features in fraud analytics.

first time, we set its recency equal to zero as is the case for AU04. Obviously, a zero or small recency could indicate anomalous behavior

In fact, the Recency feature can also be defined as follows:

$$Recency = e^{-\gamma t} \tag{1.11}$$

Here  $t$  is the time-interval between two consecutive payment transfers in which, for example, the same authentication method was used.  $\gamma$  is a user-specified parameter which is typically rather small, for example 0.02. Notice that recency is always a number between 0 and 1. Figure 1.20 shows that recency indeed decreases when the time-interval gets bigger. The parameter  $\gamma$  determines how fast the recency

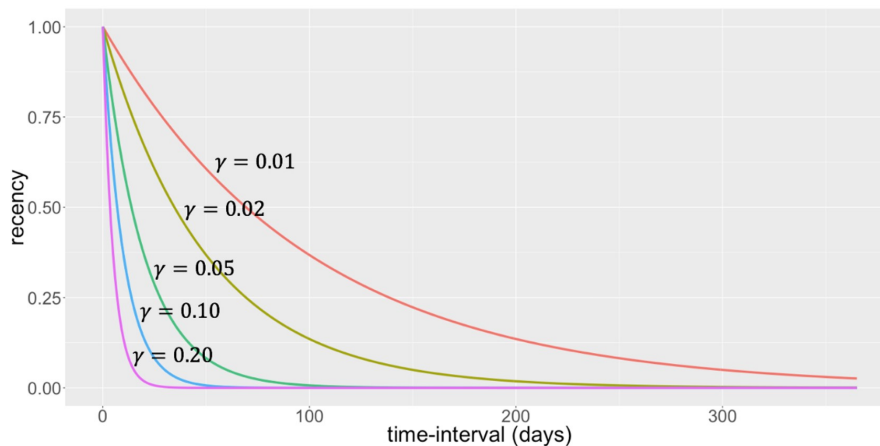


Figure 1.20: Defining the Recency feature as a negative exponential.

decreases. For larger values of  $\gamma$ , recency will decrease quicker with time and vice versa.  $\gamma$  can be chosen in various ways. If one wants to assume that Recency has to be equal to 0.01 after 180 days,  $\gamma$  becomes  $-\log(0.01)/180$ .

### 1.18.3 Domain Specific Features

Domain specific features are features that take into account the specific characteristics of a particular domain. They can be gathered by means of expert input (often originating from years of experience) or from surveys. In what follows, we give some examples of domain specific features in credit risk, fraud detection, marketing analytics, HR analytics and web analytics.

Corporate credit risk modeling typically relies on a variety of ratios measuring profitability (e.g., added value/sales, cash-flow/equity, return on equity, etc.) liquidity (e.g. current ratio, quick ratio,) and solvency

(e.g., debt/equity, debt/assets, interest coverage ratio, etc.). Developed by Edward Altman in 1968, the z-score model combines five different financial ratios using a weighted average to predict the probability of bankruptcy of a company. It is still nowadays used as a bankruptcy risk indicator in, e.g., Bloomberg and other reports and can serve as an interesting domain specific feature to help boost the performance of contemporary credit risk models. In retail credit risk models, bureau scores such as the well-known FICO score in the US, are another example. Another example is the Loan-to-Value (LTV) ratio which compares the outstanding exposure of the loan to the value of the collateral and is typically an important predictor in credit loss models estimating the loss given default (LGD).

**Altman z-model.**

The Altman z-score is a linear combination of 5 accounting ratios: Working Capital/Total Assets, Retained Earnings/Total Assets, Earnings before Interest and Taxes/Total Assets, Market or Book Value of Equity/Total Liabilities, and Net Sales/Total Assets. A higher z score reflects a more healthy firm and thus a lower bankruptcy risk. Extensions of the original z-score model have been provided for privately held and non-manufacturing firms.

In fraud detection, the usage of Benford’s law is quite popular. This law describes a fascinating fact about the distribution of the first digits of numbers. Assume you take a newspaper at a random page and write down the first or leftmost digit which is either 1, 2, 3, until 9. If all digits are equally likely then we expect to observe each digit as the first digit in approximately 1 out of 9 or 11% of cases. Benford’s law, however, predicts a different distribution for the first digit of a number. According to Benford’s law, the probability that the first digit equals 1 is about 30%, while it’s only 4.6% for digit 9. In fact, the leading digit,  $d=1$  to 9, occurs with a probability:

$$P(d) = \log_{10} \left( 1 + \frac{1}{d} \right) \tag{1.12}$$

This distribution is visualised in Figure 1.21.

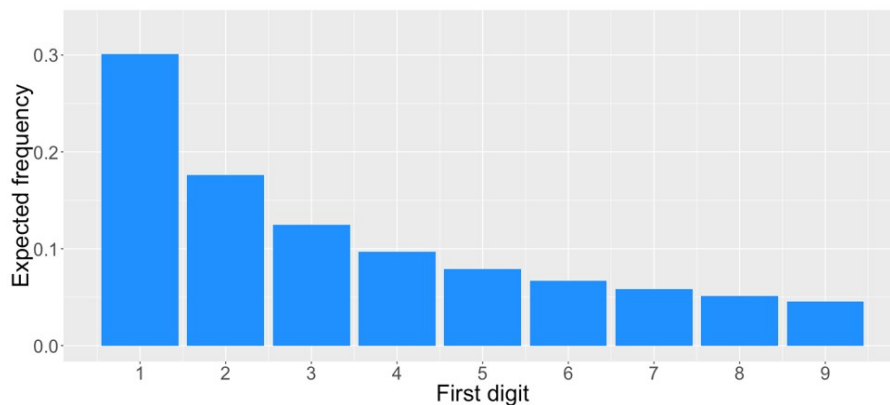


Figure 1.21: Benford’s law.

The antifraud rationale behind the use of the law is that producing empirical distributions of digits that conform to the law is difficult for non-experts. Fraudsters may thus be biased toward simpler and more intuitive distributions, such as the uniform. Strong deviations from the expected frequencies might indicate that the data is suspicious, possibly manipulated, and thus fraudulent. If Benford’s law is not satisfied, then it is probable that the involved data was manipulated and further investigation is required. Conversely, if a data set complies with Benford’s law, it can still be fraudulent. Data sets satisfying one of the following conditions typically conform to Benford’s Law:

- data where numbers represent sizes of facts or events;
- data in which numbers have no relationship to each other;
- data sets that grow exponentially or arise from multiplicative fluctuations;

- mixtures of different data sets;
- some well-known infinite integer sequences

Typically, the more orders of magnitude that the data covers (at least 4 digits) and the more observations it has (typically 1000 or more), the more likely the data set will satisfy Benford's Law. Benford's Law is even legally admissible as evidence in the US in criminal cases at the federal, state and local levels. In fact, it has been successfully used for check fraud, electricity theft, forensic accounting and payments fraud. In a fraud prediction setting, we can create predictive features based on Benford's law. More specifically, we can featurize the discrepancy between the empirical distribution and Benford's law using a statistical distance measure such as the Kullback-Leibler divergence or Kolmogorov-Smirnov statistic. These can then be added to the fraud data set for predictive modeling.

#### **Benford's law.**

Benford's law was first discovered by astronomist Newcomb in 1881 and later rediscovered by Benford in 1938. Both noted that in a book of logarithms the first pages, with low first digits, are more frequently used than the last pages with digits 7, 8 and 9 since they were more dirty. In that time logarithm tables were frequently used to speed up the multiplication of two numbers. Benford analyzed the distribution of the first digits in 20 tables concerning populations, molecular weights, mathematical sequences and death rates. In total he observed 20,229 numbers by hand.

We already discussed the RFM features above which are widely used in marketing analytics. Another popular feature is the Net Promotor Score (NPS). This is an index registered by Frederick Reichheld, Bain & Company, and Satmetrix, and was introduced by Reichheld in Reichheld [2004]. The value is obtained from customer surveys as a number between 0 and 10 answering the question: *"How likely are you to recommend the company's products/services to a relative or a friend?"*. The obtained scores can then be categorized into 3 classes: detractor (NPS between 0 and 6), neutral (NPS 7 or 8) or promoters (NPS 9 or 10). The NPS feature can be used as a variable in customer segmentation or a target variable in churn prediction.

In HR analytics, various types of features have proven to be predictive or descriptive depending upon the task at hand. As an example, a binary feature indicating if someone updated their LinkedIn profile recently is usually very predictive for upcoming employee churn. The Bradford score (BS) is a feature which is often used in employee absenteeism analysis. More specifically, it quantifies how short, frequent, unplanned absences are more disruptive to firms than longer absences and is calculated as  $BS = S^2 \times D$  with  $S$  being the total number of cases of absence in a given period and  $D$  being the total days an individual was absent in a given period. As an example an employee being twice absent for 5 days gives a worse BS of 40 compared to 10 for being only once absent for 10 days. Finally, the employee Net Promotor Score (eNPS) is an extension of the above discussed NPS score but now measuring how likely your employees are to recommend your organisation as a good place to work.

In web analytics, various features have been defined which all typically quantify customer engagement. Some popular examples are the

- bounce rate: percentage of one page site visits
- conversion rate: percentage of visits that include an action of interest such as download a product description pdf, asking for a quote, or making a purchase;
- time on site: amount of time a customer spent on the site
- pages visited: the number of pages visited during a visit
- new visitors: percentage of new incoming visitors
- return visitors: percentage of visitors that visit the site multiple times

### 1.18.4 Trend Features

Trend features are another important set of features which usually turn out to be very predictive in analytics. Trends summarize the historical evolution of a variable in various ways. Trends can be computed in an absolute or relative way as you can see illustrated in the below two formulas for the variable  $x$  measured at time  $t$  and  $t - h$

$$\frac{x_t - x_{t-h}}{h} \quad (1.13)$$

$$\frac{x_t - x_{t-h}}{x_{t-h}}$$

They can be especially useful for size variables such as asset size or loan amounts and ratios. When computing trends, it is important to consider what happens if the denominator becomes 0. Recent values can also be assigned a higher weighted. Trends can also be featurized using time series techniques, such as ARIMA or GARCH models [Hyndman and Athanasopoulos, 2014].

### 1.18.5 Transformations

A common goal of variable transformations or re-expressions is to make the distributions more symmetric, normal or improving skewness and/or kurtosis. In non-linear, additive models transformations are applied to improve model performance by introducing non-linear effects.

The logarithmic transformation transforms a variable by taking the natural logarithm of it.

$$x \rightarrow f(x) = \log(x) \quad (1.14)$$

It is often applied to size variables like loan amounts, assets sizes and gross domestic product (GDP). Size variables are usually non-negative and very often right skewed. For variables that exhibit strong right skewness and range over multiple orders of magnitude, the log transform is a benchmark transformation applied in credit risk modeling. Note that this transformation is only defined for positive values  $x \geq 0$ . In order to deal with negative values, one may shift the variable  $x$  by a fixed constant  $c$ . In practical applications, the constant  $c$  is put equal to the theoretical population minimum, the population minimum or the variable floor defined in the outlier treatment procedure.

To the left of Figure 1.22 you can see the distribution of an amount variable of a credit scoring data set. You can clearly see that the distribution is skewed to the right. In right subfigure, you can see the impact of a logarithmic transformation. You can see that the distribution of the transformed variable is more symmetric and normal.

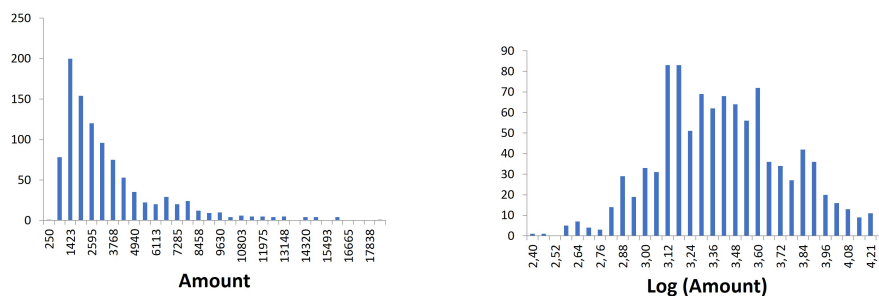


Figure 1.22: Logarithmic transformation.

The simple power transform is defined as

$$x \rightarrow f(x) = x^\lambda \quad (1.15)$$

Powers larger than 1 will make a distribution more right skewed, whereas powers smaller than 1 will make it more left skewed. The best value of  $\lambda$  is usually determined through experimentation and by visually

inspecting the distribution of the transformed variable. It can also be set by optimizing a performance measure on a validation set such as the AUC or maximum profit. As left skewed variables are rather rare, one typically applies  $\lambda < 11$ . In Figure 1.23 you can see the power transformation illustrated. To the left, you can see it for positive values of  $\lambda$ , whereas to the right you can see it for negative values of  $\lambda$ .

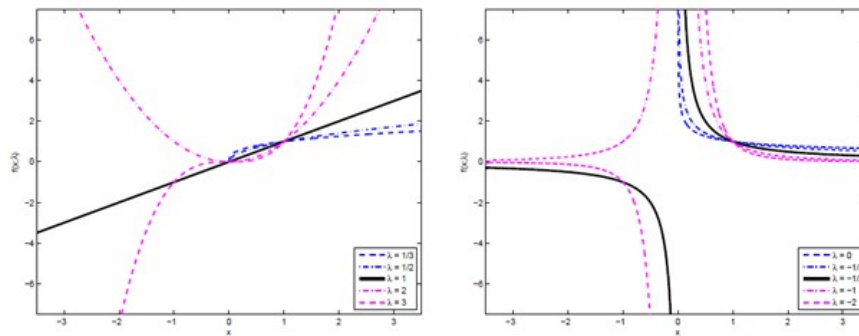


Figure 1.23: Power transformation.

The Box-Cox power transformations are a well-known type of power transformation to improve symmetry, normality or model fit. The Box-Cox transformations are defined as follows [Box and Cox, 1964]

$$x \mapsto f(x; \lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(x) & \lambda = 0. \end{cases} \quad (1.16)$$

This family of transformations is continuous in the parameter  $\lambda$ . Each transformation crosses the y-axis at  $x = 1$ . The Box-Cox transformation represents a family of power transformations that incorporates and extends many popular transformations such as the square root, cube root, natural log, reciprocal square root, reciprocal, square, . . . Note that these transformations are only defined for positive values  $x > 0$ . In order to deal with negative values, one may shift the variable  $x$  by a fixed constant  $c$ . In practical applications, the constant  $c$  is put equal to the theoretical population minimum, the population minimum or the variable floor defined in the outlier treatment procedure. Note that applying the Box-Cox power transformation is not a guarantee for normality. As discussed previously, the lambda parameter can be set by experimentation, by visually inspecting the distribution of the transformed variable, or by optimizing a performance measure on a validation set such as the AUC or maximum profit. In Figure 1.24 you can see the Box-Cox transformation illustrated. To the left you can see it for positive values of  $\lambda$ , whereas to the right you can see it for negative values of  $\lambda$ .

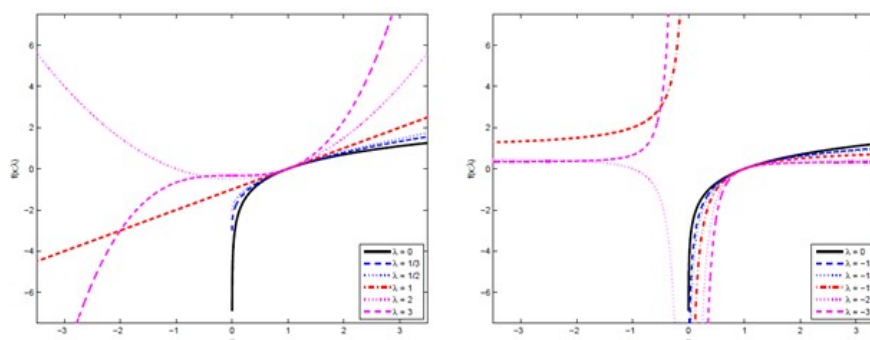


Figure 1.24: Box-Cox transformation.

The Yeo Johnson transformation is another popular transformation. It is defined as you can see here [Yeo

and Johnson, 2000]

$$x \mapsto f(x; \lambda) = \begin{cases} ((1+x)^\lambda - 1)/\lambda, & \lambda \neq 0, x \geq 0, \\ \log(x+1), & \lambda = 0, x \geq 0, \\ -((1-x)^{2-\lambda} - 1)/(2-\lambda), & \lambda \neq 2, x < 0, \\ -\log(-x+1), & \lambda = 2, x < 0. \end{cases} \quad (1.17)$$

It can be easily verified that for  $\lambda = 1$  the identity transformation is obtained. If  $\lambda = 0$ , the logarithmic transformation is applied to the positive values, whereas negative values are transformed accordingly via a smooth transition between positive and negative values. If  $\lambda = 2$ , the logarithmic transformation is applied to the negative values, whereas positive values are transformed accordingly via a smooth transition between positive and negative values. As previously, the  $\lambda$  parameter can be set by through experimentation, by visually inspecting the distribution of the transformed variable, or by optimizing a performance measure on a validation set such as the AUC or maximum profit. In Figure 1.25 you can see the Yeo-Johnson transformation illustrated. To the left you can see it for positive values of lambda, whereas to the right you can see it for negative values of  $\lambda$ .

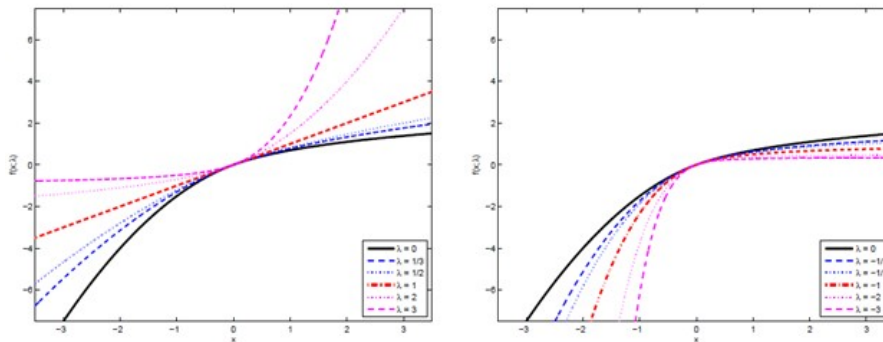


Figure 1.25: Yeo Johnson transformation.

Here you can see an example optimization procedure to determine the parameter  $\lambda$ :

- split the data into a training, validation and test set.
- Use the training set to build the model, the validation set to determine the optimal  $\lambda$  value and the test set to get an independent performance estimate.
- specify a range and step size for  $\lambda$ . For each  $\lambda$ , build a model on the training set and evaluate its performance, e.g. in terms of AUC, on the validation set.
- select the  $\lambda$  with the best validation set AUC.
- With this  $\lambda$  value, build a model on the combined training and validation set and measure its performance on the independent test set.

## 1.19 DIMENSIONALITY REDUCTION

### 1.19.1 PCA

Principal component analysis (PCA) is a data transformation technique that is aimed at reducing the dimensionality of the data by forming new variables that are linear combinations of the original variables while preserving as much variance as possible. You can see this illustrated in Figure 1.26 where sales revenue is contrasted against advertising budget. The two measurement dimensions represented by the  $X$  and  $Y$  axes do not adequately capture the actual variance that is present in the data. This is clearly situated at a 45-degree angle compared to the  $X$  and  $Y$  axes and is better captured by the two principal



components: PC1 (capturing 90% of the variance) and PC2 (capturing 7% of the variance). PC1 could be referred to as the market influence factor since it reflects the effectiveness of the advertising efforts whereas PC2 could be named the revenue residual factor modeling the effects of other influences such market conditions, competition, or external economic conditions.

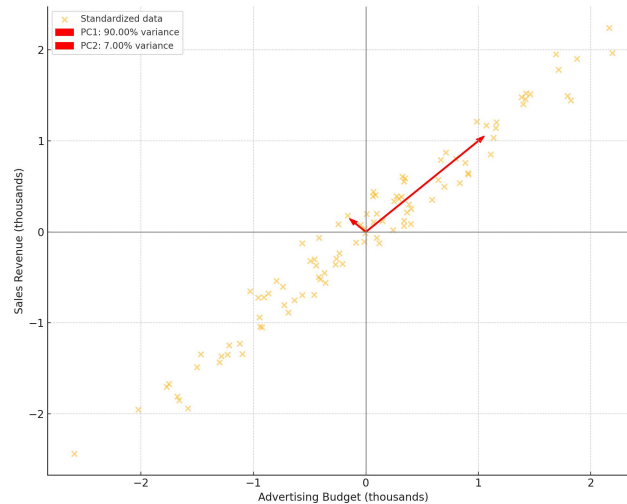


Figure 1.26: Principal Component Analysis.

PCA starts by standardizing all variables  $\mathbf{X}_{\text{standardized}} = \mathbf{X} - \text{mean}(X)$  such that each has a mean of zero. The covariance matrix  $\mathbf{\Sigma}$  of the standardized data is

$$\mathbf{\Sigma} = \frac{1}{n-1} \mathbf{X}_{\text{standardized}}^T \mathbf{X}_{\text{standardized}}$$

In a next step,  $\mathbf{\Sigma}$  is decomposed into its eigenvalues ( $\lambda$ ) and eigenvectors ( $\mathbf{v}$ ):

$$\mathbf{\Sigma} \mathbf{v} = \lambda \mathbf{v}$$

The eigenvectors represent the directions or principal components, and the eigenvalues indicate the variance in these directions. The eigenvectors can then be sorted by their corresponding eigenvalues in descending order. The largest eigenvalues correspond to the components with the most variance and hence the most informative ones. The top  $k$  can then be selected for further analysis. These eigenvectors form the principal component matrix  $\mathbf{W}$  which can then be used to transform the original data:

$$\mathbf{X}_{\text{reduced}} = \mathbf{X}_{\text{standardized}} \mathbf{W}$$

Principal component analysis is a powerful data reduction and preprocessing technique. Because the principal components are uncorrelated, they do not cause multicollinearity problems when used as inputs into a predictive model. However, its key shortcoming is that in a real-life application, the resulting PCs might be difficult to interpret because each is calculated as a linear weighted combination of the original variables. Furthermore, it assumes that the data relationships are linear, which may not capture non-linear structures in the data.

### 1.19.2 t-SNE

t-SNE stands for t-Distributed Stochastic Neighbor Embedding and was developed by van der Maaten and Hinton in [van der Maaten and Hinton, 2008]. t-SNE is also a dimensionality reduction technique, comparable to PCA. However, PCA is a linear dimension reduction technique that seeks to maximize variance and preserves large pairwise distances. In other words, things that are different end up far apart.

This can lead to suboptimal reduction with non-linear data. On the contrary, t-SNE seeks to preserve local similarities by focusing on small pairwise distances.

t-SNE is a non-linear dimensionality reduction technique based on manifold learning. It assumes that the data points lie on an embedded non-linear manifold within a higher-dimensional space. A manifold is a topological space that locally resembles a Euclidean space near each data point. Examples are a 2 dimensional manifold, locally resembling a Euclidean plane near each point, or a 3D surface which can be described by a collection of such 2 dimensional manifolds. A higher dimensional space can thus be well “embedded” in a lower dimensional space.

t-SNE works in 2 steps. In step 1, a probability distribution representing a similarity measure over pairs of high-dimensional data points is constructed. In Step 2, a similar probability distribution over the data points in the low-dimensional map is constructed. The Kullback–Leibler divergence (also known as information gain or relative entropy) between the two distributions is then minimized. Remember that the Kullback–Leibler divergence is a measure of similarity between two probability distributions.

Let’s first elaborate on step 1. We start by measuring the similarities between data points in the high dimensional space. The similarity of  $x_j$  to  $x_i$  is the conditional probability,  $p_{j|i}$  that  $x_i$  would pick  $x_j$  as its neighbor if the neighbors were picked in proportion to their probability density assuming a Gaussian centered at  $x_i$ . We then measure the density of all other data points under a Gaussian distribution and normalize to make sure the probabilities sum to 1. Hence, the corresponding probabilities become:

$$p_{j|i} = \frac{e^{-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}}} \tag{1.18}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

$$p_{ii} = 0$$

Note that  $p_{ij}$  represents the joint probabilities which can be obtained by summing both conditional probabilities and dividing by two times  $N$ .  $N$  indicates the dimensionality of the original data points. Note that because we are only interested in modeling pairwise similarities, we set the value of  $p_{i|i}$  and  $p_{ii}$  to zero. You can see this illustrated in Figure 1.27. Suppose we have three data points:  $x_i$ ,  $x_j$  and  $x_k$ . We then compute the conditional and joint probabilities using the formulas in Equation 1.18. For the 2D example shown here,  $N$  equals 2. We then define a Gaussian distribution centered at  $x_i$  and plot it as shown in Figure

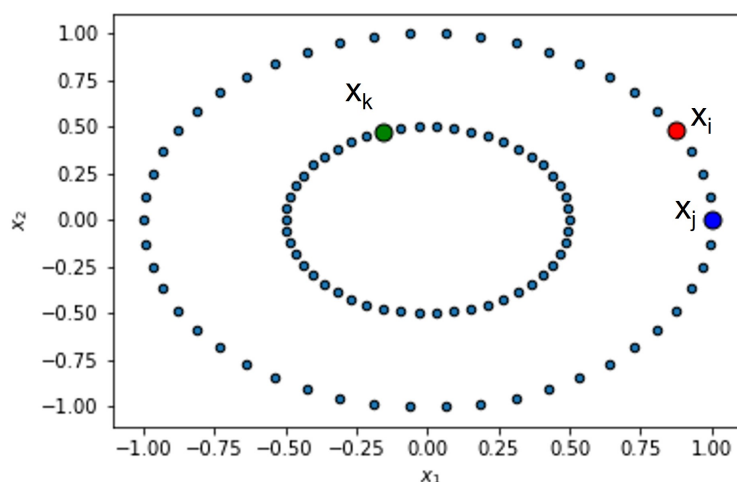


Figure 1.27: Example data for t-SNE.

1.28. The yellow area around  $x_i$  represents the center of the Gaussian distribution. We can then calculate

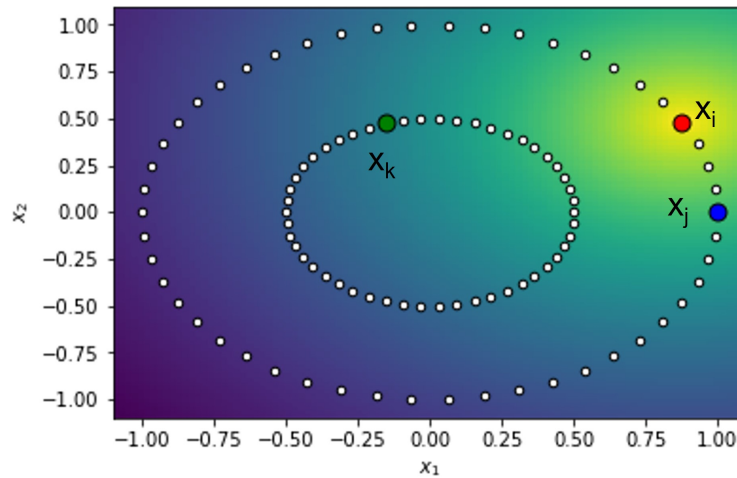


Figure 1.28: Defining a Gaussian distribution in t-SNE.

the probabilities and find that  $p_{j|i}$  equals  $0.78/z$  and  $p_{k|i}$  equals  $0.6/z$ .  $z$  represents the normalization term to make sure the probabilities sum to 1. In other words, it represents the denominator in Equation 1.18. We can then normalize the distance values for every  $k$  different from  $i$ .  $p_{j|i}$  then becomes  $0.78/55.62$  or  $0.01$ . We now compute the two joint probabilities  $p_{ij} = 0.0069$  and  $p_{ik} = 0.0049$ . It is clear that  $x_i$  and  $x_j$  are more similar than  $x_i$  and  $x_k$ .

Before we proceed to step 2, let's briefly revisit the Gaussian kernel. First, note that it uses the Euclidean distance. Furthermore, the bandwidth of the Gaussian kernel,  $\sigma$ , is data point-dependent. It is set based upon the perplexity which is a measure to estimate how well the distribution predicts a sample. Perplexity essentially reflects the effective number of close neighbors that each data point has, balancing the attention between local and global aspects of the data. A higher perplexity considers more neighbors for a more global view, while a lower perplexity focuses more on the local neighborhood. More specifically,  $\sigma_i$  is set in such a way that the perplexity of the conditional distribution equals a predefined perplexity using the bisection method. We refer to the original paper for more information about the latter [van der Maaten and Hinton, 2008]. As a result, the bandwidth parameter  $\sigma_i$  is adapted to the density of the data: smaller values are used in denser parts of the data space. This is one of the key user-specified hyperparameters of t-SNE.

In step 2 of t-SNE, we measure the similarities between the data points in the low dimensional space. In other words, we now calculate  $q_{ij}$  between the mapped data points  $y_i, y_j$ , etc. in the low dimensional space. A Student's t distribution is used to measure the similarities with the degrees of freedom equal to the dimensionality in the mapped space minus 1. The reason we use a Student's t-distribution is because it has fatter tails than a Gaussian distribution, which helps to mitigate the crowding problem by giving a higher probability to points that are moderately far apart. This distribution allows for a more effective spread of clusters in the visualization, as it more easily captures the structure of the data in lower dimensions. Again, we set  $q_{ii}$  equal to zero. Also note that, as opposed to step 1, we don't have a perplexity parameter here. We can now quantify the distances between  $p_{ij}$  and  $q_{ij}$ .  $q_{ij}$  obviously depends on the locations of the data points in the mapped space. These locations of the data points in the mapped space are determined by minimizing the Kullback-Leibler divergence as follows:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

The optimization can be performed using a standard gradient descent algorithm.

t-SNE works especially well when dealing with high dimensional data, such as images or word documents. It is important to note that t-SNE is not a clustering technique. A two-level clustering can however be easily

performed on the mapped space, using for example,  $k$ -means clustering, DBSCAN or other clustering techniques (see Chapter ??). Most implementations of t-SNE only allow the lower-dimensional space to be 2D or 3D. Essentially, t-SNE learns a non-parametric mapping. In other words, there is no explicit function that maps the data from the original input space to the map. Hence, it is not possible to embed new test points in an already existing map. This makes t-SNE good for exploratory data analysis, but less suitable as a dimensionality reduction technique in a predictive pipeline setup. Note however, that extensions exist that learn a multivariate regressor to predict the map location from the input data or construct a regressor that minimizes the t-SNE loss directly, e.g. by means of a neural network architecture. As t-SNE uses a gradient descent based approach, the general remarks regarding learning rates and initialization of mapped points apply. The initialization is sometimes done by using Principal Component Analysis, but do note that the default settings usually work well. The most important parameter is the perplexity, as we already mentioned before. The perplexity can be viewed as a knob that sets the number of effective nearest neighbors, similar to the  $k$  in  $k$ -nearest neighbor as we discuss later. As said, it depends on the density of the data. A denser data set requires a larger perplexity. Typical values range between 5 and 50. Obviously, different perplexity values can lead to very different results. You can see this illustrated in Figure 1.29. To the left, you can see the original data set. You can then see the results of applying t-SNE with perplexity values of 2, 5, 30, 50 and 100.

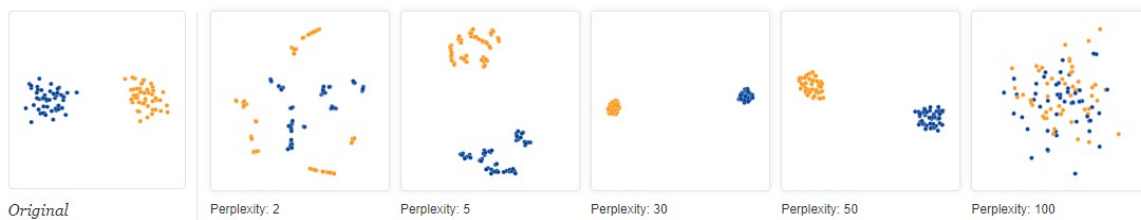


Figure 1.29: Impact of perplexity.

## 1.20 CONCLUDING REMARKS

Data preprocessing and feature engineering are essential activities during the development of analytical models. Far too often we witnessed analytical models that started from inadequate data preprocessing with obviously negative impact on model development, performance and usage. An example could be a clustering model developed on a data set without proper standardisation of the variables to a similar scale or a regression model estimated on data with outliers. Moreover, the type of data preprocessing activities considered obviously depend upon the analytical technique used. Hence, in a multi technique setup (for, e.g., model benchmarking purposes or in an ensemble setup) multiple data preprocessing and feature engineering pipelines may need to be built in parallel, each tailored to the technique at hand.

In essence, the goal of data preprocessing and feature engineering is to optimally prepare the data for analysis. It is important to note that any of the activities discussed here can be revisited after the first analytical models have been developed. As an example, the result of a decision tree or even neural network analysis may help in defining new features to improve model performance, or even add these to more simpler (e.g. regression based) models so as to boost their performance whilst maintaining their simplicity and thus model interpretability.

Part 2: Predictive Analytics

## Bibliography

- B. Baesens and A. De Caigny. *Customer Lifetime Value Modeling with Applications in Python and R*. Independently published, 2022.
- B. Baesens and S. vanden Broucke. *Practical Web Scraping for Data Science*. 2018. ISBN 978-1-4842-3581-2.
- B. Baesens and S. vanden Broucke. *Managing Model Risk*. Independently published, 2021.
- B. Baesens, V. Van Vlasselaer, and W. Verbeke. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*. SAS Institute Inc. John Wiley & Sons, Incorporated, 2015. ISBN 9781119146841. URL <https://books.google.be/books?id=daNmjwEACAAJ>.
- B. Baesens, D. Roesch, and H. Scheule. *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS*. John Wiley & Sons, 2016.
- B. Baesens, S. Höppner, and T. Verdonck. Data engineering for fraud detection. *Decision Support Systems*, 150:113492, 2021. ISSN 0167-9236. doi: <https://doi.org/10.1016/j.dss.2021.113492>. URL <https://www.sciencedirect.com/science/article/pii/S0167923621000026>. Interpretable Data Science For Decision Making.
- G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):pp. 211–252, 1964. ISSN 00359246. URL <http://www.jstor.org/stable/2984418>.
- R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014. ISBN 9780987507105. URL <https://books.google.be/books?id=gDuRBAAAQBAJ>.
- W. Lemahieu, B. Baesens, and S. vanden Broucke. *Principles of Database Management: The Practical Guide to Storing, Managing and Analyzing Big and Small Data*. Cambridge University Press, 2018. ISBN 9781107186125. URL <https://books.google.be/books?id=aeRfDwAAQBAJ>.
- H.T. Moges, K. Dejaeger, W. Lemahieu, and B. Baesens. A multidimensional analysis of data quality for credit risk management: New insights and challenges. *Information Management*, 50(1):43–58, 2013. ISSN 0378-7206. doi: <https://doi.org/10.1016/j.im.2012.10.001>. URL <https://www.sciencedirect.com/science/article/pii/S0378720612000730>.
- Frederick Reichheld. The one number you need to grow. *Harvard business review*, 81:46–54, 124, 06 2004.
- L. Thomas, D. Edelman, and J. Crook. *Credit Scoring and its Applications*. 01 2002. ISBN 0-89871-483-4.
- L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. URL <http://jmlr.org/papers/v9/vandermaaten08a.html>.

- T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke. Special issue on feature engineering editorial. *Machine Learning*, 150:113492, 2021. ISSN 0885-6125. doi: <http://doi.org/10.1007/s10994-021-06042-2>.
- R.Y. Wang and D.M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–33, 1996. ISSN 07421222. URL <http://www.jstor.org/stable/40398176>.
- In-Kwon Yeo and Richard A. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 12 2000. ISSN 0006-3444. doi: 10.1093/biomet/87.4.954. URL <https://doi.org/10.1093/biomet/87.4.954>.