

RAIN PREDICTION

A Project Work

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

**COMPUTER SCIENCE & ENGINEERING WITH SPECIALIZATION
IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

Submitted by:

PRABHJOT SINGH

20BCS6895

GAURAV SHARMA

20BCS6879

RAVI

20BCS8808

Under the Supervision of:

DR.AMIT GARG

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

PUNJAB

MAY,2022



**CHANDIGARH
UNIVERSITY**

Discover. Learn. Empower.

Annexure-4 (A typical specimen of table of contents)

Table of Contents

Title Page	i
Abstract	ii
List of Figures	iii
List of Tables (optional)	iv
Timeline / Gantt Chart	v
	v
1. INTRODUCTION*	1
1. 1.1 Problem Definition	1
2. 1.2 Project Overview/Specifications* (page-1 and 3)	2
3. 1.3 Hardware Specification	3
4. 1.4 Software Specification	4
1.3.1	4
1.3.2	
...	
2. LITERATURE SURVEY	5
2.1 Existing System	5
2.2 Proposed System	6
	7
3. PROBLEM FORMULATION	
4. RESEARCH OBJECTIVES	40
5. METHODOLOGY	47
6. TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK	
7. REFERENCES	
8. APPENDICES	

List of Tables

<i>Table Title</i>	<i>page</i>
<i>3.1 Quantities of Materials Required in the Designs with Different Grades of Concrete</i>	<i>10</i>

List of Figures

Figure Title

page

3.1	<i>Joint in a steel moment resisting frame (a) geometry, and (b) in-plane lateral distortional shear force on it. Results of analytical study (a)</i>	<i>11</i>
3.2	<i>Idealised trilinear model used in this study of or RC Frame buildings with masonry infilled walls; (b) Mean DRF spectra of Uttarkashi earthquake strong motions records derived for bare and masonry infilled RC frame buildings characteristics with $k=2,=2$, and 0.2. The spectra correspond to ductility values of 1,2,3,5,8,10,12 and 15. Dark and dashed lines correspond to bare and infilled frame buildings respectively.</i>	<i>11</i>

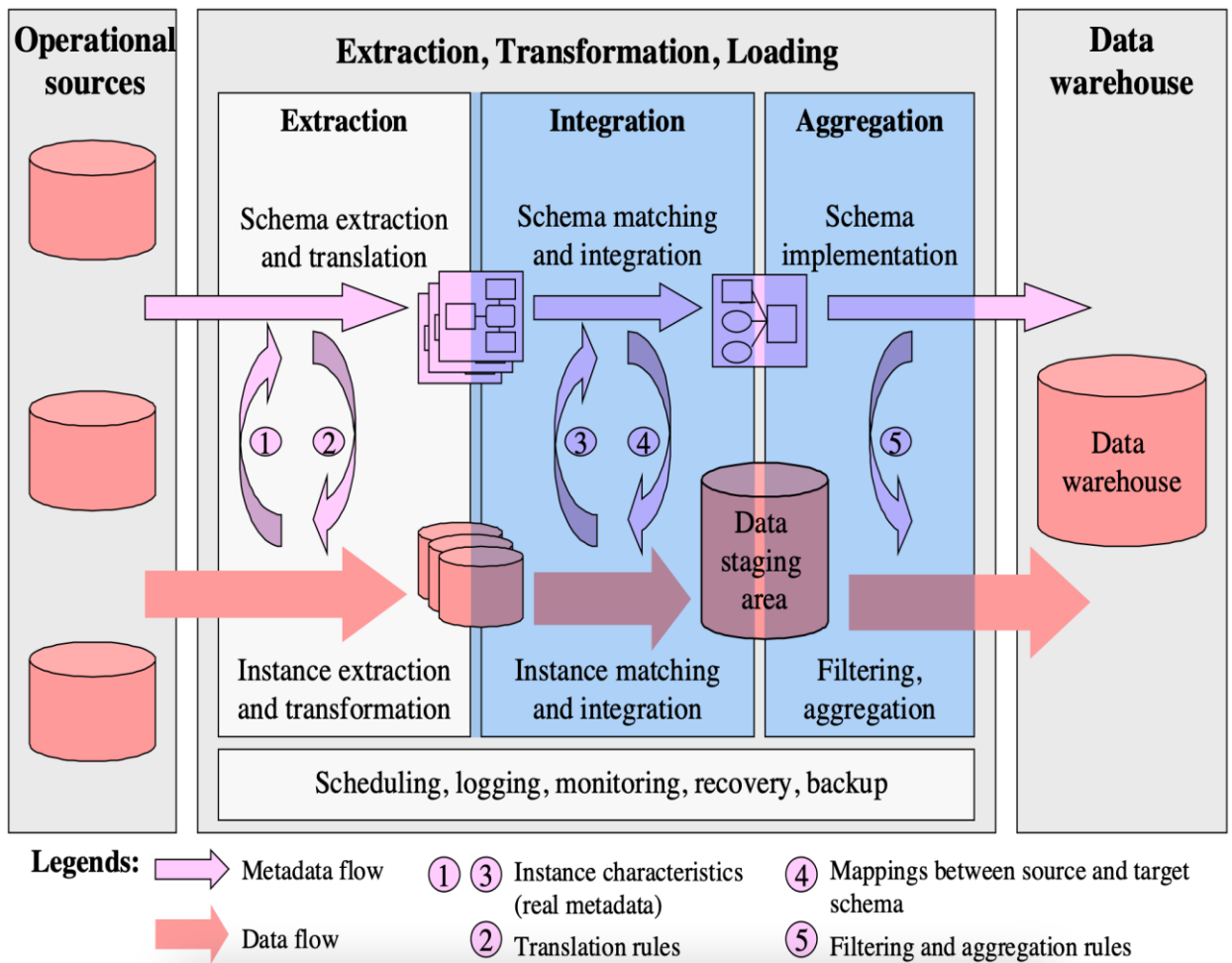
List of Symbols

<i>Symbol</i>	<i>Description</i>
----------------------	---------------------------

A_{st}	Area of steel reinforcement bars on tension face
A_{sc}	Area Of steel reinforcement bars on compression face
A_{sv}	Area of two A_{sv} legs of the closed stirrups
b	Breadth of rectangular beam section
d	Effective depth of rectangular beam section
d	Effective cover on compression face
$f_{c,ave}$	Average compressive stress in concrete
f_{sc}	Stress in steel on the compression side
f_y	Characteristic strength of steel reinforcement bars
S_v	Spacing of the S_v stirrups
x_u	Depth of neutral axis from compression face
X	Depth of centroid of the compression block in concrete
T_c	Shear strength offered by concrete

1 INTRODUCTION

Data cleaning, also called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data. Data quality problems are present in single data collections, such as files and databases, e.g., due to misspellings during data entry, missing information or other invalid data. When multiple data sources need to be integrated, e.g., in data warehouses, federated database systems or global web-based information systems, the need for data cleaning increases significantly. This is because the sources often contain redundant data in different representations. In order to provide access to accurate and consistent data, consolidation of different data representations and elimination of duplicate information become necessary.



1.1 Welcome to the Cyclistic bike-share analysis case study! In this case study, you will perform many real-world tasks of a junior data analyst. You will work for a fictional company, Cyclistic, and meet different characters and team members. In order to answer the key business questions, you will follow the steps of the data analysis process: **ask, prepare, process, analyze, share, and act**. Along the way, the **Case Study Roadmap** tables — including guiding questions and key tasks — will help you stay on the right path.

1.1.1 By the end of this lesson, you will have a portfolio-ready case study. Download the packet and reference the details of this case study anytime. Then, when you begin your

job hunt, your case study will be a tangible way to demonstrate your knowledge and skills to potential employers.

2 LITERATURE REVIEW

Kim et al. [25] proposed VUDDY, which is a scalable approach for detection of vulnerable code clones. This approach can detect vulnerabilities efficiently and accurately in large software. They able to achieve extreme level of scalability by using function-level granularity and a length-filtering techniques that decreases number of signature comparisons. Most interesting feature of this technique is that it can even detect variants of known vulnerabilities. To achieve extreme level of scalability, they used function-level granularity and length-filtering techniques to reduce number of signature comparisons.

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.

2.1 Literature Review Summary

Table 2.1: Literature review summary

Year and citation	Article Title	Purpose of the study	Tools/ Software used	Comparison of technique done	Source (Journal/ Conference)	Findings	Data set (if used)	Evaluation parameters
2010								

How do you clean data?



While the techniques used for data cleaning may vary according to the types of data your company stores, you can follow these basic steps to map out a framework for your organization.

Step 1: Remove duplicate or irrelevant observations

Remove unwanted observations from your dataset, including duplicate observations or irrelevant observations. Duplicate observations will happen most often during data collection. When you combine data sets from multiple places, scrape data, or receive data from clients or multiple departments, there are opportunities to create duplicate data. De-duplication is one of the largest areas to be considered in this process. Irrelevant observations are when you notice observations that

do not fit into the specific problem you are trying to analyze. For example, if you want to analyze data regarding millennial customers, but your dataset includes older generations, you might remove those irrelevant observations. This can make analysis more efficient and minimize distraction from your primary target—as well as creating a more manageable and more performant dataset.

Step 2: Fix structural errors

Structural errors are when you measure or transfer data and notice strange naming conventions, typos, or incorrect capitalization. These inconsistencies can cause mislabeled categories or classes. For example, you may find “N/A” and “Not Applicable” both appear, but they should be analyzed as the same category.

Step 3: Filter unwanted outliers

Often, there will be one-off observations where, at a glance, they do not appear to fit within the data you are analyzing. If you have a legitimate reason to remove an outlier, like improper data-entry, doing so will help the performance of the data you are working with. However, sometimes it is the appearance of an outlier that will prove a theory you are working on. Remember: just because an outlier exists, doesn't mean it is incorrect. This step is needed to determine the validity of that number. If an outlier proves to be irrelevant for analysis or is a mistake, consider removing it.

Step 4: Handle missing data

You can't ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered.



1. As a first option, you can drop observations that have missing values, but doing this will drop or lose information, so be mindful of this before you remove it.
2. As a second option, you can input missing values based on other observations; again, there is an opportunity to lose integrity of the data because you may be operating from assumptions and not actual observations.
3. As a third option, you might alter the way the data is used to effectively navigate null values.

Step 5: Validate and QA

At the end of the data cleaning process, you should be able to answer these questions as a part of basic validation:

- Does the data make sense?
- Does the data follow the appropriate rules for its field?
- Does it prove or disprove your working theory, or bring any insight to light?
- Can you find trends in the data to help you form your next theory?
- If not, is that because of a data quality issue?

False conclusions because of incorrect or “dirty” data can inform poor business strategy and decision-making. False conclusions can lead to an embarrassing moment in a reporting meeting when you realize your data doesn’t stand up to scrutiny. Before you get there, it is important to create a culture of quality data in your organization. To do this, you should document the tools you might use to create this culture and what data quality means to you.

Components of quality data

Determining the quality of data requires an examination of its characteristics, then weighing those characteristics according to what is most important to your organization and the application(s) for which they will be used.

5 characteristics of quality data

1. **Validity.** The degree to which your data conforms to defined business rules or constraints.
2. **Accuracy.** Ensure your data is close to the true values.
3. **Completeness.** The degree to which all required data is known.

4. **Consistency.** Ensure your data is consistent within the same dataset and/or across multiple data sets.
5. **Uniformity.** The degree to which the data is specified using the same unit of measure.

Benefits of data cleaning

Having clean data will ultimately increase overall productivity and allow for the highest quality information in your decision-making. Benefits include:

- Removal of errors when multiple sources of data are at play.
- Fewer errors make for happier clients and less-frustrated employees.
- Ability to map the different functions and what your data is intended to do.
- Monitoring errors and better reporting to see where errors are coming from, making it easier to fix incorrect or corrupt data for future applications.
- Using tools for data cleaning will make for more efficient business practices and quicker decision-making.

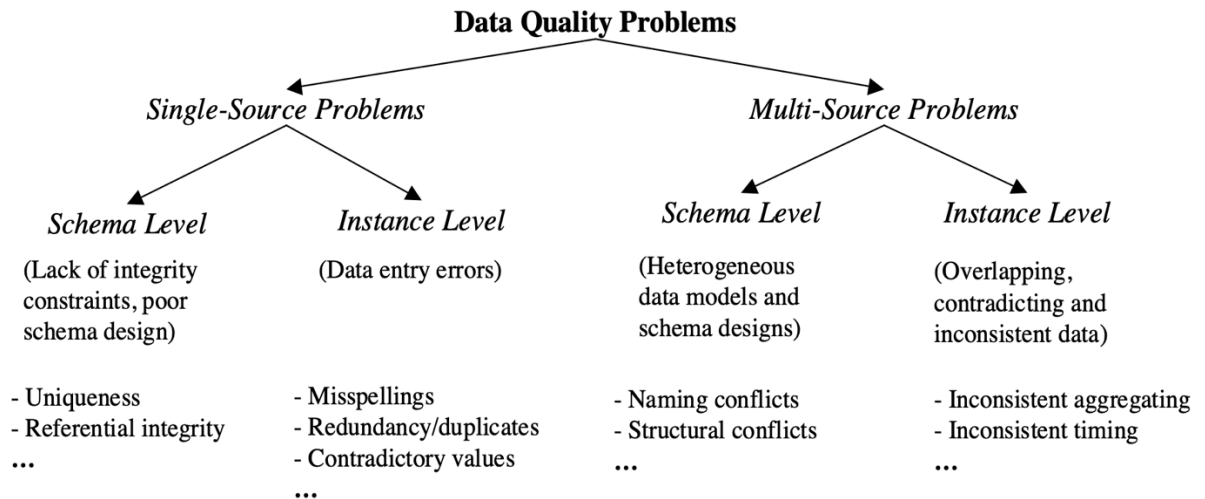
Data cleaning tools and software for efficiency

Software like Tableau Prep can help you drive a quality data culture by providing visual and direct ways to combine and clean your data. Tableau Prep has two products: Tableau Prep Builder for building your data flows and Tableau Prep Conductor for scheduling, monitoring, and managing flows across your organization. Using a data scrubbing tool can save a database administrator a significant amount of time by helping analysts or administrators start their analyses faster and have more confidence in the data. Understanding data quality and the tools you need to create, manage, and transform data is an important step toward making efficient and effective business decisions. This crucial process will further develop a data culture in your organization. To see how Tableau Prep can impact your organization, read about how marketing agency Tinuiti centralized 100-plus data sources in Tableau Prep and scaled their marketing analytics for 500 clients.

3 PROBLEM FORMULATION

During software development, clones can occur in software intentionally or unintentionally. Developers tend to clone fragments of software during development to save efforts and expedite the development process. It is more important for any organization to have the right data as compared to a large data set. Data cleansing solutions can have several problems during the process of data scrubbing. The company needs to understand the various problems and figure out how to tackle them.

From the literature review, it is observed that studies highlight the need of efficient and scalable approach for detecting code clones having software vulnerability. The existing techniques are not able to detect all types of vulnerable code clones. Different approaches suffer from high false negative rate and not scalable to large software systems due to high time complexity. So firstly, there is a need. The warehouse should contain unified data and not in a scattered manner. The data warehouse must have a documented system which is helpful for the employees to easily access the data from different sources. Data cleaning also further helps to improve the data quality by removing inaccurate data as well as corrupt and duplicate entries. Second same subject systems should be used to compare the approaches which detect



Data cleaning approaches

In general, data cleaning involves several phases

- *Data analysis*: In order to detect which kinds of errors and inconsistencies are to be removed, a detailed data analysis is required. In addition to a manual inspection of the data or data samples, analysis programs should be used to gain metadata about the data properties and detect data quality problems.
- *Definition of transformation workflow and mapping rules*: Depending on the number of data sources, their degree of heterogeneity and the “dirtyness” of the data, a

large number of data transformation and cleaning steps may have to be executed. Sometime, a schema translation is used to map sources to a common data model; for data warehouses, typically a relational representation is used. Early data cleaning steps can correct single-source instance problems and prepare the data for integration. Later steps deal with schema/data integration and cleaning multi-source instance problems, e.g., duplicates. For data warehousing, the control and data flow for these transformation and cleaning steps should be specified within a workflow that defines the ETL process (Fig. 1).

The schema-related data transformations as well as the cleaning steps should be specified by a declarative query and mapping language as far as possible, to enable automatic generation of the transformation code. In addition, it should be possible to invoke user-written cleaning code and special-purpose tools during a data transformation workflow. The transformation steps may request user feedback on data instances for which they have no built-in cleaning logic.

- *Verification:* The correctness and effectiveness of a transformation workflow and the transformation definitions should be tested and evaluated, e.g., on a sample or copy of the source data, to improve the definitions if necessary. Multiple iterations of the analysis, design and verification steps may be needed, e.g., since some errors only become apparent after applying some transformations.
- *Transformation:* Execution of the transformation steps either by running the ETL workflow for loading and

refreshing a data warehouse or during answering queries on multiple sources.

- *Backflow of cleaned data:* After (single-source) errors are removed, the cleaned data should also replace the dirty data in the original sources in order to give legacy applications the improved data too and to avoid redoing the cleaning work for future data extractions. For data warehousing, the cleaned data is available from the data staging area (Fig. 1).

The transformation process obviously requires a large amount of metadata, such as schemas, instance-level data characteristics, transformation mappings, workflow definitions, etc. For consistency, flexibility and ease of reuse, this metadata should be maintained in a DBMS-based repository [4]. To support data quality, detailed information about the transformation process is to be recorded, both in the repository and in the transformed instances, in particular information about the completeness and freshness of source data and lineage information about the origin of transformed objects and the changes applied to them. For instance, in Fig. 3, the derived table *Customers* contains the attributes *CID* and *Cno*, allowing one to trace back the source records.

In the following we describe in more detail possible approaches for data analysis (conflict detection), transformation definition and conflict resolution. For approaches to schema translation and schema integration, we refer to the literature as these problems have extensively been studied and described [2][24][26]. Name conflicts are typically resolved by renaming; structural conflicts require a partial restructuring and merging of the input schemas.

Conflict resolution

A set of transformation steps has to be specified and executed to resolve the various schema- and instance- level data quality problems that are reflected in the data sources at hand. Several types of transformations are to be performed on the individual data sources in order to deal with single-source problems and to prepare for integration with other sources. In addition to a possible schema translation, these preparatory steps typically include:

- *Extracting values from free-form attributes:* Free-form attributes often capture multiple individual values that should be extracted to achieve a more precise representation and support further cleaning steps such as instance matching and duplicate elimination. Typical examples are name and address fields (Table 2, Fig. 3, Fig. 4). Required transformations in this step are reordering of values within a field to deal with word transpositions, and value extraction for attribute splitting.
- *Validation and correction:* This step examines each source instance for data entry errors and tries to correct them automatically as far as possible. Spell checking based on dictionary lookup is useful for identifying and correcting misspellings. Furthermore, dictionaries on geographic names and zip codes help to correct address data. Attribute dependencies (birthdate – age, total price – unit price / quantity, city – phone area code,...) can be utilized to detect problems and substitute missing values or correct wrong values.
- *Standardization:* To facilitate instance matching and integration, attribute values should be converted to a

consistent and uniform format. For example, date and time entries should be brought into a specific format; names and other string data should be converted to either upper or lower case, etc. Text data may be condensed and unified by performing stemming, removing prefixes, suffixes, and stop words. Furthermore, abbreviations and encoding schemes should consistently be resolved by consulting special synonym dictionaries or applying predefined conversion rules.

Dealing with multi-source problems requires restructuring of schemas to achieve a schema integration, including steps such as splitting, merging, folding and unfolding of attributes and tables. At the instance level, conflicting representations need to be resolved and overlapping data must to be dealt with. The *duplicate elimination* task is typically performed after most other transformation and cleaning steps, especially after having cleaned single-source errors and conflicting representations. It is performed either on two cleaned sources at a time or on a single already integrated data set. Duplicate elimination requires to first identify (i.e. match) similar records concerning the same real world entity. In a second step, similar records are merged into one record containing all relevant attributes without redundancy. Furthermore, redundant records are purged. In the following we discuss the key problem of instance matching. More details on the subject are provided elsewhere in this issue [22].

In the simplest case, there is an identifying attribute or attribute combination per record that can be used for matching records, e.g., if different sources share the same primary key or if there are other common unique

attributes. Instance matching between different sources is then achieved by a standard equi-join on the identifying attribute(s). In the case of a single data set, matches can be determined by sorting on the identifying attribute and checking if neighboring records match. In both cases, efficient implementations can be achieved even for large data sets. Unfortunately, without common key attributes or in the presence of dirty data such straightforward approaches are often too restrictive. To determine most or all matches a “fuzzy matching” (approximate join) becomes necessary that finds similar records based on a matching rule, e.g., specified declaratively or implemented by a user-defined function [14][11]. For example, such a rule could state that person records are likely to correspond if name and portions of the address match. The degree of similarity between two records, often measured by a numerical value between 0 and 1, usually

7

depends on application characteristics. For instance, different attributes in a matching rule may contribute different weight to the overall degree of similarity. For string components (e.g., customer name, company name,...) exact matching and fuzzy approaches based on wildcards, character frequency, edit distance, keyboard distance and phonetic similarity (soundex) are useful [11][15][19]. More complex string matching approaches also considering abbreviations are presented in [23]. A general approach for matching both string and text data is the use of common information retrieval metrics. WHIRL represents a promising representative of this category using the cosine distance in the vector-space model

for determining the degree of similarity between text elements [7].

Determining matching instances with such an approach is typically a very expensive operation for large data sets. Calculating the similarity value for any two records implies evaluation of the matching rule on the cartesian product of the inputs. Furthermore sorting on the similarity value is needed to determine matching records covering duplicate information. All records for which the similarity value exceeds a threshold can be considered as matches, or as match candidates to be confirmed or rejected by the user. In [15] a multi-pass approach is proposed for instance matching to reduce the overhead. It is based on matching records independently on different attributes and combining the different match results. Assuming a single input file, each match pass sorts the records on a specific attribute and only tests nearby records within a certain window on whether they satisfy a predetermined matching rule. This reduces significantly the number of match rule evaluations compared to the cartesian product approach. The total set of matches is obtained by the union of the matching pairs of each pass and their transitive closure.

How to implement a successful data cleaning process

Clean data is the foundation of discovery and insight. The extreme effort your team puts forth to analyze, cultivate and visualize data is a complete waste of time if the data is dirty. Of course, dirty data isn't new. It has plagued decisions long before computers became commonplace. And now that computer technology is pervasive in everyday life, the problem has only compounded.

The first thing a company needs to determine is whether or not they have dirty data in their midst. Fortunately, this is easily done. The answer is 'Yes.' Everyone has dirty data and that means you have dirty data. Now that we're over that hurdle, the next two questions we must pose are more challenging to answer, "Which data is dirty?" and "How do we clean our data?"

Over the years I've seen many companies and teams compose a herculean effort to clean their data. They involve large dedicated teams, major project schedules, and months or years of effort. All of them have ended in much the same way, failure. At least the original task of the company's data being clean at the end of the project was a failure.

The truth is the shape of the project tends to change. It morphs from a linear project with a beginning and end, followed by an expected deliverable of clean data into a cyclical process that

will persist indefinitely. However, the ultimate goal of clean data for analysis and insights is achieved.

An example of failure...

Before I explain the steps involved in a successful cyclical process for cleaning data, let's take a moment to explore the reasons gargantuan data cleaning projects like the example presented above will always fail. The linear project approach to cleaning data has an inherent assumption leading us to repeated failure. It assumes the data you are cleaning is somewhat static in nature. New data will enter your system, but the new data will be entered correctly and not contain errors or be dirty. But if dirty data is somehow introduced into the source systems again, the processes you have put in place account for all possible forms of dirty data issues that could come your way in the future.

This is not a realistic picture of how source systems work. For example, let's say we have software used by the human resources department. It allows the HR staff to enter employee names and track their progress through all the on-boarding requirements from initial date of hire to the employee being fully trained for their position.

This software has been written (or at least customized) by a 3rd party software development team your company used to tailor the software to its particular on-boarding processes. The software saves all data into a SQL transactional database, as

we would expect. One of the fields in the database stores each employee's score for each of their assessments throughout the on-boarding process. Of course, if a particular assessment has not been completed, no score would be entered. Your initial review of the data reveals the incomplete assessments contain an empty string in the score field whereas the completed assessments contain an actual score value in the field.

Your team implements cleaning transformations to ensure these empty string values are changed into NULL values in your analytics data tables so that you can easily find and ignore them during your calculations. Problem solved! Or so you think...

Months after your data cleaning efforts are complete, one of the HR employees finds an issue in the HR software itself and they raise the concern with the software development team. The 3rd party team chooses to resolve the problem by changing the behavior of their software to no longer use an empty string value for incomplete assessment scores, but they now use a zero as the incomplete placeholder. At first, this seems to solve the problems for HR, but soon their reports from the analysts begin reflecting extremely low score averages across the onboarding groups.

When your team is finally asked to investigate, you realize the issue comes from the averaging methods being used. Since empty strings were initially the incomplete indicators and you converted them to NULLS, the average of the scores could be

completed with a simple average function because it ignores NULLS. It does not ignore zeros. Therefore, the averages are now including the zeros from the incomplete assessments.

Back to the drawing board!

A cyclical cleaning process will work

While the above example is a simple one and real life problems are much more complex, it clearly demonstrates the problems inherent with linear data cleaning approaches.

Instead, we constantly change and add to the processes affecting the data throughout our business and therefore, we are continually changing our circumstances introducing new opportunities for problematic data. And to make matters worse, we not only change the processes, systems, sources, fields, and numerous other elements involved with the collection of data, we are adding new data at an ever increasing rate.

By the time you've created a method for cleaning the data and implemented it, many or all of these factors have changed — many times over.

The linear data cleaning path is doomed to failure and should be abandoned. Instead, let's begin with a systemized version of the cyclical process everyone ends up adopting in the end.

Automate the Cleaning Steps you just Implemented

In a hospital, once the patient has been treated and is considered well enough for discharge, they still may have follow-up appointments or tasks they need to perform like taking medication. With data cleaning, multiple follow-up tasks may also be necessary, but the one follow-up task that is always required involves automation.

No matter what your company's change process is, the one thing you must do is make the changes persistent through automation. Automation comes in many forms. You may need to change the existing ETL process or introduce an automated process that cleans the data post ETL. Automation can be achieved through any language or system that works for you and your company: SQL, Python, C#, SAS, and the list goes on. A common automation system for companies using Microsoft products is SQL Server Integration Services (or SSIS). The scheduled execution of these tasks can be as simple as cron or Microsoft Task Manager or SQL Agent. It doesn't necessarily need to be sophisticated. But it needs to be automated.

If you allow the cleaning process to remain manual, you will very quickly overwhelm your team with recurring manual work and hope of taking on new data cleaning efforts will be forfeited.

I've said this elsewhere, but the quickest way to render your team useless is to overwhelm them with recurring manual work. All exploratory, cleaning and initial data wrangling work is manual. But once the process is cleanly defined, it must be automated if your team has any hope of continuing to impact your company's insights and decisions.

Repeat

Now that you've taken the cleaning process all the way through analysis, identifying, assessing, prioritizing, team assignment, establishing a cleaning process, and automating the cleaning process — now it's time to repeat the process. Analysts will confirm the result of your team's work and be thankful, but they will also send you new “data” patients and the process starts again for these new patients who are ill and need your healing touch.

A final reminder to formally establish your cyclical data cleaning process...

The doctors and nurses saving lives in the hospital emergency room aren't just winging it. They've been training for years. They can practically perform their work in their sleep because it has been ingrained in them through hours of formalized training. Not only have they been diligently trained in medical procedures and knowledge, the policies and procedures used in the ER to triage a patient, admit a patient, prioritize a patient, all the way through treating the patient have been thoroughly studied and formalized in order to give any patient coming through their doors the best chance of survival.

Medical personnel and hospital administrators know that a strong, formalized plan leaves much less chance of error and that leads to greater success. The exact same benefits from formalizing a plan are true for data cleaning.

As outlined in the opening of this article, large linear based projects for data cleaning typically end up with this same cyclical process in the end. The linear process failed them and because the work must still be accomplished, the team tackles each problem as it comes to light. But many times arriving at this same process after a failed linear attempt leaves the process in an informal state. It's really just performed in an ad hoc manner.

Avoid this pitfall!

Without formalization, the process outcomes will ebb and flow between success and failure. The successful cleaning of each newly discovered dirty data issue will always be up for grabs. Your team will work inconsistently and their motivation for the job will come and go. Even if your ad hoc approach begins to work well over time as your team gels, it's upended every time a team member leaves or a new member is hired. And here's why...there's no plan to follow.

Some days your team will knock it out of the park when they are feeling good about what they do. But other days the recurrent work and recollection of failure from the initial data cleaning project will push them to question their work and the nature of their job.

“Why can't we create a process to fix all of this instead of constantly facing a fire every day?”

“Why doesn't management care that we are already overworked? They just keep sending us more requests.”

“When will we ever catch up?”

If you will take the time and give the effort to formalize the cyclical process of cleaning your data, your team will have a

roadmap to follow and the other departments in your organization will have a guide to properly interact with your team. In many ways, it's just perspective, but the formalization of the process removes the ambiguity and gives purpose to the work being done. Formalizing is a necessary step to ensuring a consistently successful outcome to data cleaning for your team.

Business data cleansing - invention or necessity?

Nowadays, IT systems generate and process countless amounts of data daily. From a technological point of view, this is not a big challenge for our computers, servers or cloud solutions. With such a huge amount of information, the challenge is something else: the hygiene of databases, i.e. maintaining them in excellent quality.

How great this challenge is shown by the data from the latest report by Experian *2021 Global Data Management Research*:

- companies estimate that approximately $\frac{1}{3}$ of all business data about customers and potential customers is inaccurate,
- **55%** of leaders do not trust the data their organizations own,
- only **50%** believe that their CRM / ERP data is clean data and can be fully used.

Moreover, as many as **95%** of companies notice negative effects related to low data quality.

Data cleansing in 5 steps (with examples)

Different data types require a different approach, so the techniques used to clean up data may differ slightly depending on the database you are dealing with. Nevertheless, usually the business customer databases are quite similar (they always contain company registration numbers, e-mails, addresses, etc.). Hence, in the remainder of this article, we will primarily focus on data cleansing these types of records.

B2B data cleansing is a process that usually consists of at least five steps. Those are:

1. Data validation
2. Formatting data to a common value (standardization / consistency)
3. Cleaning up duplicates

4. Filling missing data vs. erasing incomplete data

5. Detecting conflicts in the database

Below we describe how data cleaning looks like in each of the stage, together with simple examples of implementation.

Data cleansing Step 1: Data Validation

Any company that has business records in its database, i.e. company data, knows perfectly that many of them is data that should be (and can be) checked for its correctness. Of course, we could assume that all company identification numbers, postal codes or e-mail addresses have been entered correctly in the database or that a business register in which we have verified the contractor certainly does not contain errors, but in practice it is not. Erroneous data can happen even in the best public commercial registers and it is no different in internal databases, where records are entered manually by employees.

This is why data validation, i.e. data verification in terms of meeting certain top-down conditions and logical principles, is the first stage of database hygiene.

For example, let's take the validation of the list of tax numbers of Polish companies imported from some X system:

6312609607	7393208668
828 131 62 12	6422105641
7392954381	PL6331813071
4980117337	5422449707
7431641598	5260300292
7393029517	583-101-48-98
744-15-16-966	7792081703
5711503502	PL 6391747857
7540335340	9691227069
5422698451	7491899923
5541007223	0000000000
984-00-78-782	754 033 53 40

If you do not have experience in working with company data, you may not know that the last digit of each tax identification number is not accidental in many of countries. In Poland this is called a 'check digit' and it is calculated on the basis of an algorithm that can be validated. Briefly, validation of the Polish check digit consists in multiplying each of the first nine digits of the tax number by weights (in sequence: 6, 5, 7, 2, 3, 4, 5, 6, 7), summing the results of this multiplication, and then dividing checksum by 11. The remainder of the division should be identical to the last digit in the tax number.

If we calculate the checksums for the tax numbers given above, it turns out that three of them are incorrect: 4980117337, 5260300292, 0000000000. Therefore, they should be deleted from the database.

This step of database cleaning (validation) has passed the following tax numbers:

828 131 62 12	7393208668
7392954381	6422105641
7431641598	PL6331813071
7393029517	5422449707
744-15-16-966	583-101-48-98
5711503502	7792081703
7540335340	PL 6391747857
5422698451	9691227069
5541007223	7491899923
984-00-78-782	754 033 53 40

Data cleansing Step 2: Formatting data to a common form

The next step in improving the quality of the database is to normalize the data to a uniform form. This procedure is used primarily to facilitate the search for information about a given company in the database.

In the table we pasted above, you can see immediately that some tax numbers were written with dashes, spaces or the prefix “PL” which stands for Poland. So now you need to format all company tax numbers to a common form. How? First of all, since we know that this is a database of Polish business clients, we can safely omit the prefix with the country code. Second, the best option in this case will be to write all numbers without any special characters separating the digits.

Thus, we get the following result:

6312609607	7393208668
8281316212	6422105641
7392954381	6331813071
7431641598	5422449707
7393029517	5831014898
7441516966	7792081703
5711503502	6391747857
7540335340	9691227069
5422698451	7491899923
5541007223	7540335340
9840078782	

Numbers are not the only values that we can bring to a consistent form in this way. E-mail addresses or website addresses can also be brought to a common form by writing all of them in lowercase. And it is certainly worth it, because what would database hygiene be if it did not make the database more consistent and easier to use? Exactly!

Data cleansing Step 3: Cleaning up duplicates

After standardizing the data format, the next step in data cleaning is to check whether our database has some duplicates that could not be detected earlier due to a different save format.

After conducting such an analysis, we discover that in our original database it was possible to find two records with the same tax number: 7540335340 and 754 033 53 40.

Our table, after removing duplicates from it, looks as follows:

6312609607	9840078782
8281316212	7393208668
7392954381	6422105641
7431641598	6331813071
7393029517	5422449707
7441516966	5831014898
5711503502	7792081703
7540335340	6391747857
5422698451	9691227069
5541007223	7491899923

The above example is limited to finding duplicates by values in one column. In practice, however, some data defines a unique record with more data arranged in different columns. For example, you can search for duplicated people by first name and last name, and in this case use two separate columns - one for the first name and the other for the last name.

Data cleansing Step 4: Filling missing data vs. erasing incomplete data

The next step in database hygiene is preventing the possession of incomplete data. Anyone who works with data at least a little knows well that the information, in addition to being reliable and up-to-date, should also be complete. Incomplete data contaminates the database, lowering its business quality.

As an example, let's take the database of B2B contractors addresses, which are saved in CRM in the following format: voivodship, commune, postal code, city and street.

VOIVODESHIP	DISTRICT	POST CODE	CITY	STREET
warmińsko-mazurskie	ostróda	14-100	ostróda	hurtowa
pomorskie	pruszcz gdański	83-031	łęgowo	ul. marco polo
lubuskie	m. gorzów wielkopolski	66-400	gorzów wielkopolski	wybiekiego
warmińsko-mazurskie		10-228	olsztyn	ul. zamenhofa
wielkopolskie	poznań-stare miasto	61-859	poznań	grobla
warmińsko-mazurskie	działdowo	14-120	dąbrówno	agrestowa
warmińsko-mazurskie		10-508	olsztyn	mickiewicza
wielkopolskie	olsztyn	10-450	olsztyn	ul. piłsudskiego
warmińsko-mazurskie	lubomino	11-135	lubomino	kopernika
wielkopolskie	grodzisk wielkopolski	62-065	grodzisk wielkopolski	ul. fabryczna
lubuskie	lubsko	68-300	lubsko	ul. włókniarzy
	m. lublin	20-327	lublin	wrońska
podkarpackie	świlcza	36-072	świlcza	ul. ---
podkarpackie	m. rzeszów	35-510	rzeszów	ślusarczyka
śląskie	dąbrowa górnicza	41-300	dąbrowa górnicza	ul. wiejska (baza pec)
pomorskie	m. gdynia	81-212	gdynia	hutnicza
wielkopolskie	poznań-stare miasto	61-623	poznań	ul. wilczak
	poznań-grunwald	60-318	poznań	ul. władysława węgorka
warmińsko-mazurskie	kętrzyn	11-400	kętrzyn	ul. rynkowa
	m. szczecin	70-556	szczecin	null

Let's assume that in our system we want to have only complete company addresses, i.e. complete data sets (incomplete data does not contribute anything to the business process). We can approach this topic in two ways:

1. **delete all records** that have an empty value in any field (which is not an ideal solution, because we lose a lot of information),
2. **complete incomplete records** (which is a much better choice, considering that a voivodeship or a commune can be easily completed based on the name of the city or postal code), and only what cannot be retrieved with a supplement (in this case, e.g. sets with empty street info) remove.

Of course, we decide to clean the database the second way.

In order to facilitate this task and perform it fully professionally, it is necessary to define some repetitive and exhaustive rules that will apply to this data set in turn. They take the following form:

- If the *voivodship* field is empty, we complete it based on the city.
- If the *city* field is empty, we check whether we can determine the city name based on the postcode field (we will not always be able to do this - there are many common postal codes for various smaller towns and villages).

- If the *commune/district* field is empty, we complete it based on the city and postal code.
- We are introducing a few rules for clearing the data in the street column, such as clearing null strings or removing values where there are no letters other than street.
- In the last step, we get rid of the records that are still left with empty values in any of the fields of a single dataset.

After applying the above set of rules, our cleaned database of company addresses looks like this:

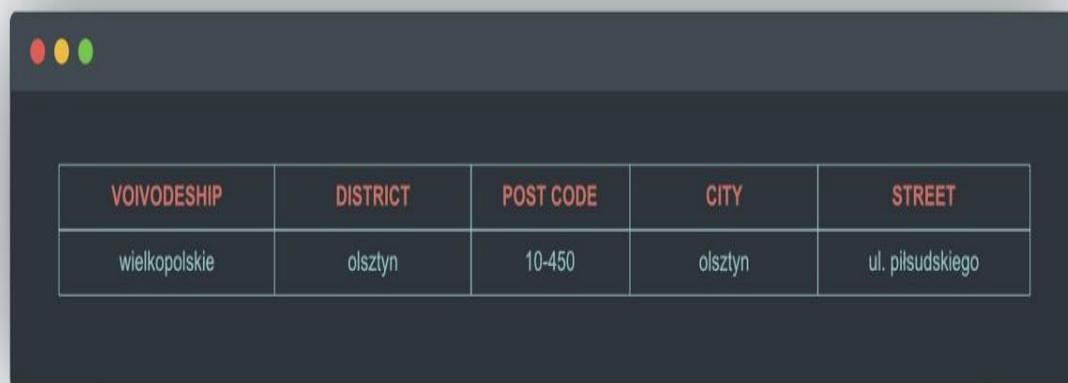


VOIVODESHIP	DISTRICT	POST CODE	CITY	STREET
warmińsko-mazurskie	ostróda	14-100	ostróda	hurtowa
pomorskie	pruszcz gdański	83-031	łęgowo	ul. marco polo
lubuskie	m. gorzów wielkopolski	66-400	gorzów wielkopolski	wybickiego
warmińsko-mazurskie	olsztyn	10-228	olsztyn	ul. zamenhofa
wielkopolskie	poznań-stare miasto	61-859	poznań	grobla
warmińsko-mazurskie	działdowo	14-120	dąbrówno	agrestowa
warmińsko-mazurskie	olsztyn	10-508	olsztyn	mickiewicza
wielkopolskie	olsztyn	10-450	olsztyn	ul. piłsudskiego
warmińsko-mazurskie	lubomino	11-135	lubomino	kopernika
wielkopolskie	grodzisk wielkopolski	62-065	grodzisk wielkopolski	ul. fabryczna
lubuskie	lubsko	68-300	lubsko	ul. włókniarzy
lubelskie	m. lublin	20-327	lublin	wrońska
podkarpackie	m. rzeszów	35-510	rzeszów	ślusarczyka
śląskie	dąbrowa górnicza	41-300	dąbrowa górnicza	ul. wiejska (baza pec)
pomorskie	m. gdynia	81-212	gdynia	hutnicza
wielkopolskie	poznań-stare miasto	61-623	poznań	ul. wilczak
wielkopolskie	poznań-grunwald	60-318	poznań	ul. władysława węgorka
warmińsko-mazurskie	kętrzyn	11-400	kętrzyn	ul. rynkowa

Data cleansing Step 5: Detecting conflicts in the database

The last step in our data quality improvement process is the so-called conflict detection. In the terminology of working with data, conflicts are data that are contradictory or mutually exclusive. As you can easily guess, properly performed data hygiene aims to track them all down and mark them properly.

Continuing the example with the address database, we can check, for example, whether the zip code, city and commune match the voivodship entered or whether there is a conflict somewhere. Performing such a quick analysis, you will notice that one of the records is incorrect:



VOIVODESHIP	DISTRICT	POST CODE	CITY	STREET
wielkopolskie	olsztyn	10-450	olsztyn	ul. piłsudskiego

In this dataset, the voivodeship does not match the rest of the address provided.

What can be done now with such a conflict? If you know who entered the data into the system, contact that person to explain the error and enter the correct values. However, if it is impossible for some reason, you should first of all properly mark this record in the database. Thanks to this, in the future it will be easier for us to decide whether to use such a record or not in further data processing. Thanks to this, if we wanted to carry out a statistical survey by provinces, for example, we would be able to simply omit such conflicting, "uncertain" records, so as not to introduce errors in the calculations.

Sometimes, the data hygiene of B2B records databases containing company numbers includes one more activity that involves the detection of conflicts, and is aimed at checking the validity of the information: namely, detecting conflicts in data with other national business registers (verifying whether a given company has an active business status in the National Court Register). Then, companies that have been deleted or suspended in the registers are appropriately marked so that we can decide later whether we want to remove them from the database.

Data hygiene, how often to do it?

The data hygiene of our clients 'and potential clients' business databases is not a topic that we can leave alone. No good

manager should assume in advance that employees of various departments have never made and will not make mistakes when entering new data or that everyone will adhere to uniform recording standards. Error is a human thing, so it simply has to be time for data cleansing in the enterprise. Either it should be performed by a properly trained employee (data analyst / programmer with knowledge of the specifics of working with data), or we should outsource this task to an external company specializing in this subject, preferably one that has an ISO / IEC 27001 information security certificate.

How often should data hygiene be carried out in the company? Well, it depends on the size of the base. Medium and large enterprises with a large number of records should repeat data cleaning every 3-6 months. For smaller companies, it is enough to do data hygiene about once a year.

4 OBJECTIVES

The proposed work is aimed to carry out work leading to the development of an approach. The proposed aim will be achieved by dividing the work into following objectives: marketers are generating a large portion of poor-quality leads, including those with improper formatting and even inaccuracies. Bad prospect information can have negative consequences, including wasted media investment, squandered resources and poor customer experience, which marketers simply can't afford."

Your dealership's CRM system represents a considerable financial outlay. Cleaning your CRM system's data will allow you to see an optimal return on your investment. It's estimated that 11.7 of people moved each year. It's easy to see what that means in terms of how quickly data will become outdated over a relatively short time. If you don't clean your database, over time, it will become filled with old, outdated customer data. According to Experian, the average business loses 12 percent of its revenue due to dirty data, and this figure doesn't take missed opportunities into account.

1. Direct Mail Costs: Despite the popularity of email and social media campaigns, direct mail marketing is still an important component of a dealership's overall marketing strategy. Spending money to print and ship mailers that the prospect never receives because of a dirty database is a waste of resources.

2. Email Campaign Performance: While there are no printing and shipping costs associated with emailing customers and prospects, spending money on an email campaign that will never reach the intended targets will drag down your email campaign's performance. With a clean database, you'll reach more customers and potential customers, which translates into more sales.

3. New Opportunities: New vehicle sales and leases, pre-owned vehicle sales, service and finance all represent opportunities for revenue. The dirtier your database, the less chance there is for your dealership to take advantage of these opportunities.

4. Ongoing Engagement: Your marketing campaigns are an important way to keep customers engaged, which creates an ongoing relationship with your dealership.

5 METHODOLOGY

The following methodology will be followed to achieve the objectives defined for proposed research work:

All data sources potentially include errors and missing values – data cleaning addresses these anomalies. Not cleaning data can lead to a range of problems, including linking errors, model misspecification, errors in parameter estimation and incorrect analysis leading users to draw false conclusions. The impact of these problems is magnified in the S-DWH environment¹ due to the planned re-use of data: if the data contain untreated anomalies, the problems will repeat. The other key data cleaning requirement in a S-DWH is storage of data before cleaning and after every stage of cleaning, and complete metadata on any data cleaning actions applied to the data. The main data cleaning processes are editing, validation and imputation. Editing and validation are sometimes used synonymously – in this manual we distinguish them as editing

describing the identification of errors, and validation their correction. The remaining process, imputation, is the replacement of missing values.

Different data types have distinct issues regards data cleaning, so data-specific processing needs to be built into a S-DWH.

- Census data – although census data do not usually contain a high percentage of anomalies, the sheer volume of responses, allied with the number of questions, so data cleaning needs to be automatic wherever possible
- Sample survey data – business surveys generally have less responses, more variables, and more anomalies than social surveys – and are more complex due to the continuous nature of the variables (compared to categorical variables for social surveys) – so data cleaning needs to be very differently defined for business and social surveys
- Administrative data – traditional data cleaning techniques do not work for administrative data due to the size of the datasets and the underlying data collection (which legally and/or practically precludes recontact to validate responses), so data cleaning needs to be automatic wherever possible.

2 Editing

Data editing can take place at different levels, and use different methods – the choice is known as the data editing strategy. Different data editing strategies are needed for each data type – there is no “one size fits all” solution for a S-DWH. Macro- and micro-editing Editing can be at the micro level, editing individual records, or the macro level, editing (aggregate) outputs.

- Macro-editing is generally subjective – eye-balling the output, in isolation and/or relative to similar outputs/previous time periods, or calculating measures of growth and applying rules of thumb to decide whether they are realistic or not. This type of editing would not suit the S-DWH environment, as outputs are separated by two layers from inputs, and given the philosophy of re-use of data it would be difficult to define a process where “the needs of the one (output) outweigh the needs of the many”. Hence nothing more is said about these methods.
- Micro-editing methods are numerous and well-established, and are appropriate for a S-DWH where editing should only take place in the sources layer. Hence these are the focus here.

Hard and soft edits

Editing methods – known as rules – detect errors, but once a response fails the treatment varies dependent on the rule type.

- Hard edits (some validity, consistency, logical and statistical) do not require validation and can be treated automatically – see below.

- Soft edits (all remaining) require external validation – see section 3.

Automatic editing

Automatic editing, mentioned in section 1 as a key option for census data, is also commonly used for business survey data as a cost- and burden-saving measure when responses fail hard edits. Given the high costs associated with development of a S-DWH, automatic editing should be implemented wherever possible – at least during initial development. However, another advantage of automatic editing applies both during development and beyond – it will lead to more timely data, as there will be less time spent validating failures, which will benefit all dependent outputs.

Selective editing

Selective (also significance) editing, like automatic editing, is a cost- and burden-saving measure. It reduces the amount of overall validation required by automatically treating the least important edit rule failures as if they were not failures – the remaining edit rule failures are sent for validation.

Validation

Data validation takes place once responses fail edit rules, and are not treated automatically. The process involves human intervention to decide on the most appropriate treatment for each failure – based on three sources of information (in priority order):

- primary – answer given during a telephone call querying the response, or additional written information (eg the respondent verified the response when recontacted)
- secondary – previous responses from the same respondent (eg if the current response, although a failure, follows the same pattern as previous responses then the response would be confirmed)
- tertiary – current responses from similar respondents (eg if there are more than one respondents in a household, their information could explain the response that failed the edit rule)

In addition to these objective sources of information, there is also a valuable subjective source – the experience of the staff validating the data (eg historical knowledge of the reasons for failures).

In a S-DWH environment, the requirement for clean data needs to be balanced against the demand for timely data. This is a motivation for automatic editing, and is also a consideration for failures that cannot be automatically treated. The process would be more objective than outside a S-DWH, as the experience of staff working on a particular data source – the subjective information source for validation – would be lost given generic teams would validate all sources. This lack

of experience could also mean that the secondary information source for validation – recognition of patterns over time – would also be less likely to be effective. This means that in a S-DWH, validation would be more likely to depend on the primary and tertiary sources of information – direct contact with respondents, and proxy information provided by similar respondents (or provided by the same respondent to another survey or administrative source).

Imputation

The final stage of data cleaning is imputation for partial missing response (item non-response) – the solution for total missing response (unit non-response) is estimation (see 1.3). To determine what imputation method to use requires understanding of the nature of the missing data.

Types of missingness Missing data can be characterized as 3 types:

- MCAR (missing completely at random) – the missing responses are a random subsample of the overall sample

- MAR (missing at random) – the rate of missingness varies between identifiable groups, but within these groups the missing responses are MCAR
- NMAR (not missing at random) – the rate of missingness varies between identifiable groups, and within these groups the probability of being missing depends on the outcome variable In a S-DWH environment, the ability to determine the type of missingness is in theory diminished due to the multiple groups and outcome variables the data could be used for, but in practice the type of missingness should be determined in terms of the primary purpose of the data source, as again it is impossible to predict all secondary uses.

Imputation methods

There is an intrinsic link between imputation and automatic editing: imputation methods define how to automatically replace a missing response based on an imputation rule; automatic editing defines how to automatically impute for a response failing an edit rule. Thus imputation methods are akin to automatic editing treatments, but the names are different.

There are a huge number of possible imputation methods – the choice is based on:

- the type of missingness – generally deterministic for MCAR, stochastic for MAR, deductive for NMAR
- testing each method against the truth – achieved by imputing existing responses, and measuring how close they imputed response is to the real response

In a S-DWH environment, the choice of imputation method should be determined based on the primary purpose of the data source – in concordance with the type of missingness. This chosen method, and its associated variance, must form part of the detailed metadata for each imputed response to ensure proper inference from all subsequent uses.

6 TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

CHAPTER 1: INTRODUCTION

Welcome to the Cyclistic bike-share analysis case study! In this case study, you will perform many real-world tasks of a junior data analyst. You will work for a fictional company, Cyclistic, and meet different characters and team members. In order to answer the key business questions, you will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act. Along the way, the Case Study Roadmap tables including guiding questions and key tasks — will help you stay on the right path —.

By the end of this lesson, you will have a portfolio-ready case study. Download the packet and reference the details of this case study anytime. Then, when you begin your job hunt, your case study will be a tangible way to demonstrate your knowledge and skills to potential employers.

CHAPTER 2: LITERATURE REVIEW

Scenario

This chapter include the literature available for you are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations. The findings of the researchers will be highlighted which will become basis of current implementation.

CHAPTER 2: BACKGROUND OF PROPOSED METHOD

This chapter will provide introduction to the concepts which are necessary to understand the proposed system.

- **Cyclistic:** A bike-share program that features more than 5,800 bicycles and 600 docking stations. Cyclistic sets itself apart by also offering reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike. The majority of riders opt for traditional bikes; about 8% of riders use the assistive options. Cyclistic users are more likely to ride for leisure, but about 30% use them to commute to work each day.
- **Lily Moreno:** The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.
- **Cyclistic marketing analytics team:** A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic's mission and business goals — as well as how you, as a junior data analyst, can help Cyclistic achieve them.
- **Cyclistic executive team:** The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

CHAPTER 4: METHODOLOGY

This chapter will cover the technical details of the proposed approach.

The following methodology will be followed to achieve the objectives defined for proposed research work:

All data sources potentially include errors and missing values – data cleaning addresses these anomalies. Not cleaning data can lead to a range of problems, including linking errors, model misspecification, errors in parameter estimation and incorrect analysis leading users to draw false conclusions. The impact of these problems is magnified in the S-DWH

environment¹ due to the planned re-use of data: if the data contain untreated anomalies, the problems will repeat. The other key data cleaning requirement in a S-DWH is storage of data before cleaning and after every stage of cleaning, and complete metadata on any data cleaning actions applied to the data. The main data cleaning processes are editing, validation and imputation. Editing and validation are sometimes used synonymously – in this manual we distinguish them as editing describing the identification of errors, and validation their correction. The remaining process, imputation, is the replacement of missing values.

Different data types have distinct issues regards data cleaning, so data-specific processing needs to be built into a S-DWH.

- Census data – although census data do not usually contain a high percentage of anomalies, the sheer volume of responses, allied with the number of questions, so data cleaning needs to be automatic wherever possible
- Sample survey data – business surveys generally have less responses, more variables, and more anomalies than social surveys – and are more complex due to the continuous nature of the variables (compared to categorical variables for social surveys) – so data cleaning needs to be very differently defined for business and social surveys
- Administrative data – traditional data cleaning techniques do not work for administrative data due to the size of the datasets and the underlying data collection (which legally and/or practically precludes recontact to validate responses), so data cleaning needs to be automatic wherever possible.

2 Editing

Data editing can take place at different levels, and use different methods – the choice is known as the data editing strategy. Different data editing strategies are needed for each data type – there is no “one size fits all” solution for a S-DWH. Macro- and micro-editing Editing can be at the micro level, editing individual records, or the macro level, editing (aggregate) outputs.

- Macro-editing is generally subjective – eye-balling the output, in isolation and/or relative to similar outputs/previous time periods, or calculating measures of growth and applying rules of thumb to decide whether they are realistic or not. This type of editing would not suit the S-DWH environment, as outputs are separated by two layers from inputs, and given the philosophy of re-use of data it would be difficult to define a process where “the needs of the one (output) outweigh the needs of the many”. Hence nothing more is said about these methods.
- Micro-editing methods are numerous and well-established, and are appropriate for a S-DWH where editing should only take place in the sources layer. Hence these are the focus here.

Hard and soft edits

Editing methods – known as rules – detect errors, but once a response fails the treatment varies dependent on the rule type.

- Hard edits (some validity, consistency, logical and statistical) do not require validation and can be treated automatically – see below.
- Soft edits (all remaining) require external validation – see section 3.

Automatic editing

Automatic editing, mentioned in section 1 as a key option for census data, is also commonly used for business survey data as a cost- and burden-saving measure when responses fail hard edits. Given the high costs associated with development of a S-DWH, automatic editing should be implemented wherever possible – at least during initial development. However, another advantage of automatic editing applies both during development and beyond – it will lead to more timely data, as there will be less time spent validating failures, which will benefit all dependent outputs.

Selective editing

Selective (also significance) editing, like automatic editing, is a cost- and burden-saving measure. It reduces the amount of overall validation required by automatically treating the least important edit rule failures as if they were not failures – the remaining edit rule failures are sent for validation.

Validation

Data validation takes place once responses fail edit rules, and are not treated automatically. The process involves human intervention to decide on the most appropriate treatment for each failure – based on three sources of information (in priority order):

- primary – answer given during a telephone call querying the response, or additional written information (eg the respondent verified the response when recontacted)
- secondary – previous responses from the same respondent (eg if the current response, although a failure, follows the same pattern as previous responses then the response would be confirmed)
- tertiary – current responses from similar respondents (eg if there are more than one respondents in a household, their information could explain the response that failed the edit rule)

In addition to these objective sources of information, there is also a valuable subjective source – the experience of the staff validating the data (eg historical knowledge of the reasons for failures).

In a S-DWH environment, the requirement for clean data needs to be balanced against the demand for timely data. This is a motivation for automatic editing, and is also a consideration for failures that cannot be automatically treated. The process would be more objective than outside a S-DWH, as the experience of staff working on a particular data source

– the subjective information source for validation – would be lost given generic teams would validate all sources. This lack of experience could also mean that the secondary information source for validation – recognition of patterns over time – would also be less likely to be effective. This means that in a S-DWH, validation would be more likely to depend on the primary and tertiary sources of information – direct contact with respondents, and proxy information provided by similar respondents (or provided by the same respondent to another survey or administrative source).

Imputation

The final stage of data cleaning is imputation for partial missing response (item non-response) – the solution for total missing response (unit non-response) is estimation (see 1.3). To determine what imputation method to use requires understanding of the nature of the missing data.

Types of missingness Missing data can be characterized as 3 types:

- MCAR (missing completely at random) – the missing responses are a random subsample of the overall sample
- MAR (missing at random) – the rate of missingness varies between identifiable groups, but within these groups the missing responses are MCAR
- NMAR (not missing at random) – the rate of missingness varies between identifiable groups, and within these groups the probability of being missing depends on the outcome variable In a S-DWH environment, the ability to determine the type of missingness is in theory diminished due to the multiple groups and outcome variables the data could be used for, but in practice the type of missingness should be determined in terms of the primary purpose of the data source, as again it is impossible to predict all secondary uses.

Imputation methods

There is an intrinsic link between imputation and automatic editing: imputation methods define how to automatically replace a missing response based on an imputation rule; automatic editing defines how to automatically impute for a response failing an edit rule. Thus imputation methods are akin to automatic editing treatments, but the names are different.

There are a huge number of possible imputation methods – the choice is based on:

- the type of missingness – generally deterministic for MCAR, stochastic for MAR, deductive for NMAR
- testing each method against the truth – achieved by imputing existing responses, and measuring how close they imputed response is to the real response

In a S-DWH environment, the choice of imputation method should be determined based on the primary purpose of the data source – in concordance with the type of missingness. This chosen method, and its associated variance, must form part of the detailed metadata for each imputed response to ensure proper inference from all subsequent uses.

CHAPTER 5: EXPERIMENTAL SETUP

This chapter will provide information about the subject system and tools used for evaluation of proposed method.

CHAPTER 6: RESULTS AND DISCUSSION

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.

Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.

Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

Ask

Three questions will guide the future marketing program:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

Moreno has assigned you the first question to answer: How do annual members and casual riders use Cyclistic bikes differently?

You will produce a report with the following deliverables:

1. A clear statement of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of your analysis
5. Supporting visualizations and key findings
6. Your top three recommendations based on your analysis

Use the following Case Study Roadmap as a guide. Note: Completing this case study within a week is a good goal.

Case Study Roadmap - Ask
Guiding questions <ul style="list-style-type: none">• What is the problem you are trying to solve?• How can your insights drive business decisions?
Key tasks <ol style="list-style-type: none">1. Identify the business task2. Consider key stakeholders
Deliverable <ul style="list-style-type: none"><input type="checkbox"/> A clear statement of the business task

Analyze

Now that your data is stored appropriately and has been prepared for analysis, start putting it to work. Use the following Roadmap as a guide:

Guiding questions <ul style="list-style-type: none"> • How should you organize your data to perform analysis on it? • Has your data been properly formatted? • What surprises did you discover in the data? • What trends or relationships did you find in the data? • How will these insights help answer your business questions?
Key tasks <ol style="list-style-type: none"> 1. Aggregate your data so it's useful and accessible. 2. Organize and format your data. 3. Perform calculations. 4. Identify trends and relationships.
Deliverable <ul style="list-style-type: none"> <input type="checkbox"/> A summary of your analysis

Follow these steps for using spreadsheets

Open your spreadsheet application, then complete the following steps:

1. Where relevant, make columns consistent and combine them into a single worksheet.
2. Clean and transform your data to prepare for analysis.
3. Conduct descriptive analysis.
4. Run a few calculations in one file to get a better sense of the data layout. Options:
 - Calculate the mean of ride_length
 - Calculate the max ride_length
 - Calculate the mode of day_of_week
5. Create a pivot table to quickly calculate and visualize the data. Options:
 - Calculate the average ride_length for members and casual riders. Try rows = member_casual; Values = Average of ride_length.
 - Calculate the average ride_length for users by day_of_week. Try columns = day_of_week; Rows = member_casual; Values = Average of ride_length.

- ● Calculate the number of rides for users by day_of_week by adding Count of trip_id to Values.
6. Open another file and perform the same descriptive analysis steps. Explore different seasons to make some initial observations.
 7. Once you have spent some time working with the individual spreadsheets, merge them into a full-year view. Do this with the tool you have chosen to use to perform your final analysis, either a spreadsheet, a database and SQL, or R Studio.
 8. Export a summary file for further analysis.

Share

Now that you have performed your analysis and gained some insights into your data, create visualizations to share your findings. Moreno has reminded you that they should be sophisticated and polished in order to effectively communicate to the executive team.

Follow these steps:

1. Take out a piece of paper and a pen and sketch some ideas for how you will visualize the data.
2. Once you choose a visual form, open your tool of choice to create your visualization. Use a presentation software, such as PowerPoint or Google Slides; your spreadsheet program; Tableau; or R.
3. Create your data visualization, remembering that contrast should be used to draw your audience's attention to the

most important insights. Use artistic principles including size, color, and shape.

4. Ensure clear meaning through the proper use of common elements, such as headlines, subtitles, and labels.
5. Refine your data visualization by applying deep attention to detail.
- 6.

Follow these steps for using SQL

Open your SQL tool of choice, then complete the following steps:

1. Import your data.
2. Explore your data, perhaps looking at the total number of rows, distinct values, maximum, minimum, or mean values.
3. Where relevant, use JOIN statements to combine your relevant data into one table.
4. Create summary statistics.
5. Investigate interesting trends and save that information to a table.

Follow these steps for using R

Open R Studio and use this script to complete the following steps:

1. Import your data.
2. Make columns consistent and merge them into a single dataframe.
3. Clean up and add data to prepare for analysis.
4. Conduct descriptive analysis.

5. Export a summary file for further analysis.

The result of proposed technique will be discussed in this chapter.

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

You will use Cyclistic's historical trip data to analyze and identify trends. Download the previous 12 months of Cyclistic

trip data here. (Note: The datasets have a different name because Cyclistic is a fictional company. For the purposes of this case study, the datasets are appropriate and will enable you to answer the business questions. The data has been made available by Motivate International Inc. under this license.) This is public data that you can use to explore how different customer types are using Cyclistic bikes. But note that data-privacy issues prohibit you from using riders' personally identifiable information. This means that you won't be able to connect pass purchases to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes.

Download the previous 12 months of Cyclistic trip data.

1) Unzip the files.

2) Create a folder on your desktop or Drive to house the files. Use appropriate file-naming conventions.

3) Create subfolders for the .CSV file and the .XLS or Sheets file so that you have a copy of the original data. Move the downloaded files to the appropriate subfolder.

4) Follow these instructions for either Excel (a) or Google Sheets (b):

A) Launch Excel, open each file, and choose to Save As an Excel Workbook file. Put it in the subfolder you created for .XLS files.

B) Open each .CSV file in Google Sheets and save it to the appropriate subfolder.

5) Open your spreadsheet and create a column called “ride_length.” Calculate the length of each ride by subtracting the column “started_at” from the column “ended_at” (for example, =D2-C2) and format as HH:MM:SS using Format > Cells > Time > 37:30:55.

6) Create a column called “day_of_week,” and calculate the day of the week that each ride started using the “WEEKDAY” command (for example, =WEEKDAY(C2,1)) in each file. Format as General or as a number with no decimals, noting that 1 = Sunday and 7 = Saturday.

7) Proceed to the analyze step.

The major finding of the work will be presented in this chapter. Also directions for extending the current study will be discussed.

Wrap-up

Congratulations on finishing the Cyclistic bike-share case study! If you like, complete one of the other case studies to continue growing your portfolio. Or, use the steps from the **ask, prepare, process, analyze, share, and act** Case Study Roadmap to create a new project all your own. Best of luck on your job search!

Our Project Coding Screenshots and execute codes:

+ Code + Text

✓ RAM 
Disk 

Editing

```
✓ [1] import numpy as np  
import pandas as pd
```

```
✓ [2] from google.colab import files
```

```
uploaded = files.upload()
```

Choose Files weatherAUS.csv

- weatherAUS.csv(text/csv) - 14094055 bytes, last modified: 12/11/2020 - 100% done

Saving weatherAUS.csv to weatherAUS.csv

```
✓ [3] dataset = pd.read_csv('weatherAUS.csv')  
X = dataset.iloc[:, [1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]].values  
Y = dataset.iloc[:, -1].values
```

```
✓ [4] print(X)
```

```
[[ 'Albury' 13.4 22.8      16.8 21.8 'No']
```

✓
36

[4] `print(X)`

```
[['Albury' 13.4 22.9 ... 16.9 21.8 'No']  
 ['Albury'  7.4 25.1 ... 17.2 24.3 'No']  
 ['Albury' 12.9 25.7 ... 21.0 23.2 'No']  
 ...  
 ['Uluru'  5.4 26.9 ... 12.5 26.1 'No']  
 ['Uluru'  7.8 27.0 ... 15.1 26.0 'No']  
 ['Uluru' 14.9 nan ... 15.0 20.9 'No']]
```

✓
37

[5] `print(Y)`

```
['No' 'No' 'No' ... 'No' 'No' nan]
```

✓
38

[6] `Y = Y.reshape(-1,1)`

✓
39

```
[7] from sklearn.impute import SimpleImputer  
imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')  
X = imputer.fit_transform(X)  
Y = imputer.fit_transform(Y)
```

```
[8] print(X)
```

```
[[ 'Albury' 13.4 22.9 ... 16.9 21.8 'No']  
 [ 'Albury'  7.4 25.1 ... 17.2 24.3 'No']  
 [ 'Albury' 12.9 25.7 ... 21.0 23.2 'No']  
 ...  
 [ 'Uluru'  5.4 26.9 ... 12.5 26.1 'No']  
 [ 'Uluru'  7.8 27.0 ... 15.1 26.0 'No']  
 [ 'Uluru' 14.9 20.0 ... 15.0 20.9 'No']]
```

```
[9] print(Y)
```

```
[[ 'No']  
 [ 'No']  
 [ 'No']  
 ...  
 [ 'No']  
 [ 'No']  
 [ 'No']]
```

```
✓ [10] from sklearn.preprocessing import LabelEncoder
      le1 = LabelEncoder()
      X[:,0] = le1.fit_transform(X[:,0])
      le2 = LabelEncoder()
      X[:,4] = le2.fit_transform(X[:,4])
      le3 = LabelEncoder()
      X[:,6] = le3.fit_transform(X[:,6])
      le4 = LabelEncoder()
      X[:,7] = le4.fit_transform(X[:,7])
      le5 = LabelEncoder()
      X[:, -1] = le5.fit_transform(X[:, -1])
      le6 = LabelEncoder()
      Y[:, -1] = le6.fit_transform(Y[:, -1])
```

```
✓ [11] print(X)
```

```
[[2 13.4 22.9 ... 16.9 21.8 0]
 [2  7.4 25.1 ... 17.2 24.3 0]
 [2 12.9 25.7 ... 21.0 23.2 0]
 ...
 [41  5.4 26.9 ... 12.5 26.1 0]
 [41  7.8 27.0 ... 15.1 26.0 0]
 [41 14.9 20.0 ... 15.0 20.9 0]]
```

✓ [12] `print(Y)`

```
[[0]
 [0]
 [0]
 ...
 [0]
 [0]
 [0]]
```

✓ [13] `Y = np.array(Y,dtype=float)`
`print(Y)`

```
[[0.]
 [0.]
 [0.]
 ...
 [0.]
 [0.]
 [0.]]
```

✓ [14] `from sklearn.preprocessing import StandardScaler`
`sc = StandardScaler()`
`X = sc.fit_transform(X)`

```
✓ [15] print(X)
```

```
[[ -1.53166617  0.19132753 -0.04135977 ... -0.01407077  0.02310362
   -0.52979545]
 [ -1.53166617 -0.75105231  0.26874452 ...  0.03244663  0.387799
   -0.52979545]
 [ -1.53166617  0.11279588  0.35331842 ...  0.62166712  0.22733303
   -0.52979545]
 ...
 [ 1.20928479 -1.06517892  0.52246622 ... -0.69632607  0.65037966
   -0.52979545]
 [ 1.20928479 -0.68822699  0.53656187 ... -0.29317521  0.63579185
   -0.52979545]
 [ 1.20928479  0.42692249 -0.45013361 ... -0.30868102 -0.10818671
   -0.52979545]]
```

```
✓ [16] from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=0)
```



```
✓ [17] print(X_train)
[[ 0.22535368  1.03946939  0.07140543 ...  0.68369032  0.08145488
   -0.52979545]
 [ 1.42012717 -0.45263203  0.11369237 ... -0.41722163  0.22733303
   -0.52979545]
 [ 0.50647685 -0.20133073 -0.14002932 ... -0.06058818 -0.02065982
    1.88752093]
 ...
 [ 1.0687232  0.75675544  0.93124006 ...  1.10234698  1.07342629
   -0.52979545]
 [ 0.57675765 -0.04426743 -0.16822062 ...  0.01694083 -0.28324049
    1.88752093]
 [ 1.63096955 -0.0285611  -0.91529006 ... -0.35519842 -0.76463838
   -0.52979545]]
```

```
✓ [18] print(Y_train)
[[1.]
 [0.]
 [0.]
 ...
 [0.]
 [0.]
 [0.]]
```

```
[19] from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=100,random_state=0)
classifier.fit(X_train,Y_train)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please use a scalar instead, or use (array([value]), dtype=[object]) to pass a scalar as a single element of a 1d array.
RandomForestClassifier(random_state=0)
```

```
[20] classifier.score(X_train,Y_train)
```

```
0.99999312525780283
```

```
[21] y_pred = le6.inverse_transform(np.array(classifier.predict(X_test),dtype=int))
Y_test = le6.inverse_transform(np.array(Y_test,dtype=int))
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:154: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please use a scalar instead, or use (array([value]), dtype=[object]) to pass a scalar as a single element of a 1d array.
y = column_or_1d(y, warn=True)
```

```
[22] print(y_pred)
```

```
['No' 'No' 'No' ... 'No' 'No' 'No']
```

```
[23] print(Y_test)
```

```
['Yes' 'Yes' 'No' ... 'Yes' 'No' 'No']
```

```
[24] y_pred = y_pred.reshape(-1,1)
```

```
Y_test = Y_test.reshape(-1,1)
```

```
[25] df = np.concatenate((Y_test,y_pred),axis=1)
```

```
dataframe = pd.DataFrame(df,columns=['Rain on Tommorrow','Predition of Rain'])
```

```
[26] print(dataframe)
```

	Rain on Tommorrow	Predition of Rain
0	Yes	No
1	Yes	No
2	No	No
3	No	Yes
4	No	No
...
29087	No	Yes
29088	No	No
29089	Yes	No
29090	No	No
29091	No	No

```
[29092 rows x 2 columns]
```



```
from sklearn.metrics import accuracy_score  
accuracy_score(Y_test,y_pred)
```

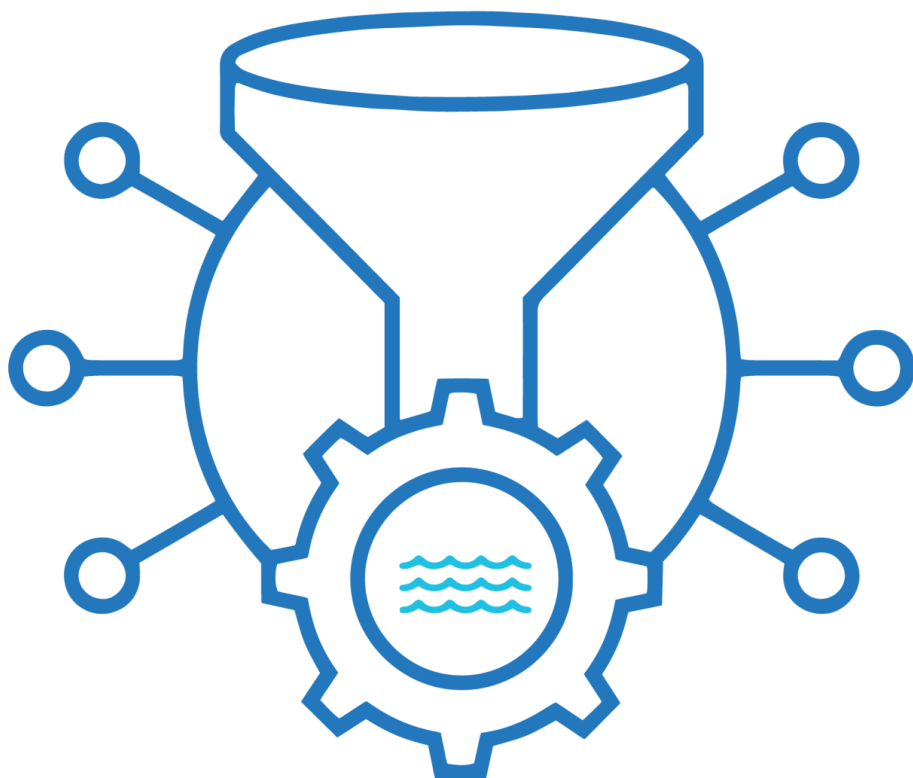
```
0.8521930427608965
```

What is data cleaning, cleansing and scrubbing?

Clean data is crucial for insightful data analysis. Data cleansing, data cleaning or data scrubbing is the first step in the overall data preparation process. It is the process of analyzing, identifying and correcting messy, raw data. Data cleaning involves filling in missing values, identifying and fixing errors and determining if all the information is in the right rows and columns. When analyzing organizational data to make strategic decisions you must start with a thorough data cleansing process. Cleaning data is crucial to data analysis. Data cleaning lays the groundwork for efficient, accurate and effective data analysis. Without cleaning data beforehand, the analysis process won't be clear or as accurate because the information in the dataset will be unorganized and scattered. Good analysis rests on clean data—it's as simple as that.

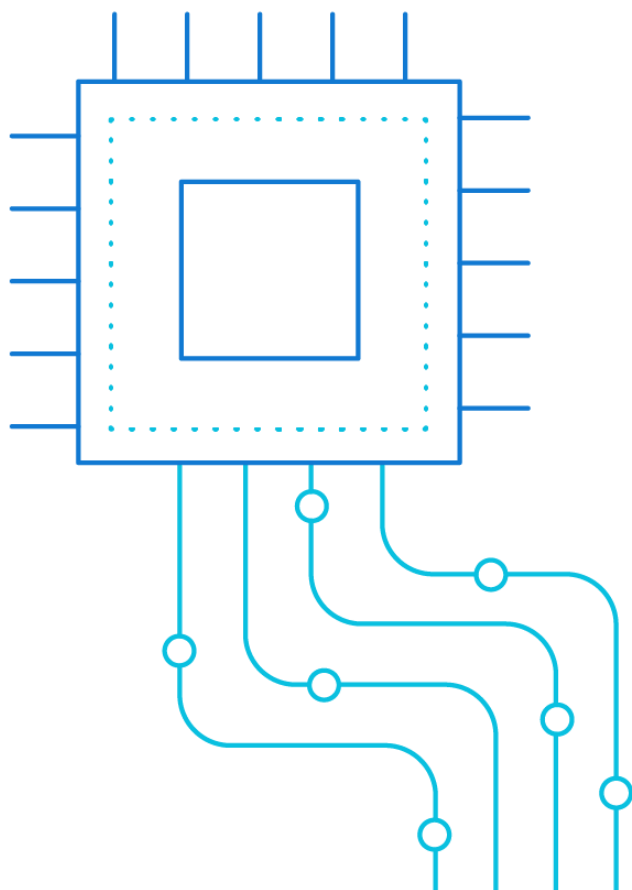
The challenges with data cleaning

Because good analysis relies on adequate data cleaning, analysts may face challenges with the data cleaning process. All too often organizations lack the attention and resources needed to perform data scrubbing to have an effect on the end result of analysis. Inadequate data cleansing and data preparation frequently allow inaccuracies to slip through the cracks. The lack of data scrubbing leading to inaccuracies is not the fault of the data analyst, but a symptom of a much larger problem of manual and siloed data cleansing and data preparation. Beyond the lackluster and faulty analysis, the larger issue with traditional data cleansing and preparation is the amount of time it takes—Forrester Research reports that up to 80% of an analyst's time is spent on data cleansing and preparation. With so much time spent scrubbing data, it's understandable why data cleaning steps are sometimes skipped over. Most organizations need a data cleaning solution that will help with analysis but reduce the time and resources spent on preparation.



How to clean up data: data scrubbing made easier

Data clean up can be difficult, but the solution doesn't need to be. Data cleaning tools make the process simpler. We have created a new approach to data preparation that helps organizations get the most value out of their data with proper data scrubbing. With its visual, user-friendly interface, Trifacta's data wrangling software allows non-technical users to wrangle data and scrub data of all shapes and sizes for sophisticated analysis. Trifacta empowers non-technical or business users to do more with their data by guiding them through the process using intelligent suggestions powered by machine learning. What was once the daunting and overwhelming task of data cleansing, is now made simple with Trifacta. Now data scrubbing won't consume valuable time, and fewer inaccuracies can slip through the cracks.

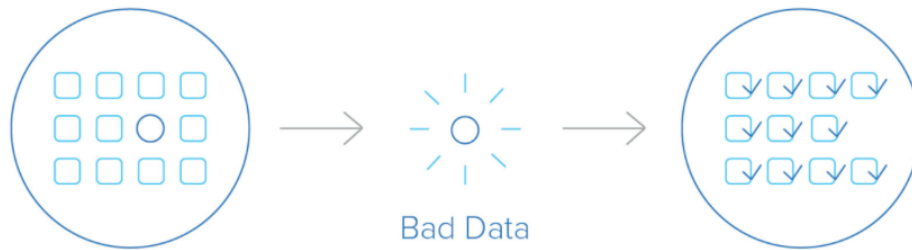


Trifacta's unique approach to data cleansing: The six-step wrangling process

Our six-step wrangling process lends itself to a more iterative data cleansing and data wrangling, ultimately leading to a more accurate analysis. The steps involved include:

1. Discovering helps the user understand what's in the data and how it can be used effectively for analysis
2. Structuring makes working with data of all shapes and sizes easy by formatting the data to be used in traditional applications
3. Cleaning or data scrubbing involves removing data that may distort your analysis or standardizing your data into a single format
4. Enriching allows the user to augment the data with internal or third-party data to enhance the data for better analysis
5. Validating brings data quality and inconsistency issues to the surface so the appropriate transformations can be applied
6. Publishing allows the user to deliver the output of your data and load into downstream systems for analysis.

Data Cleansing

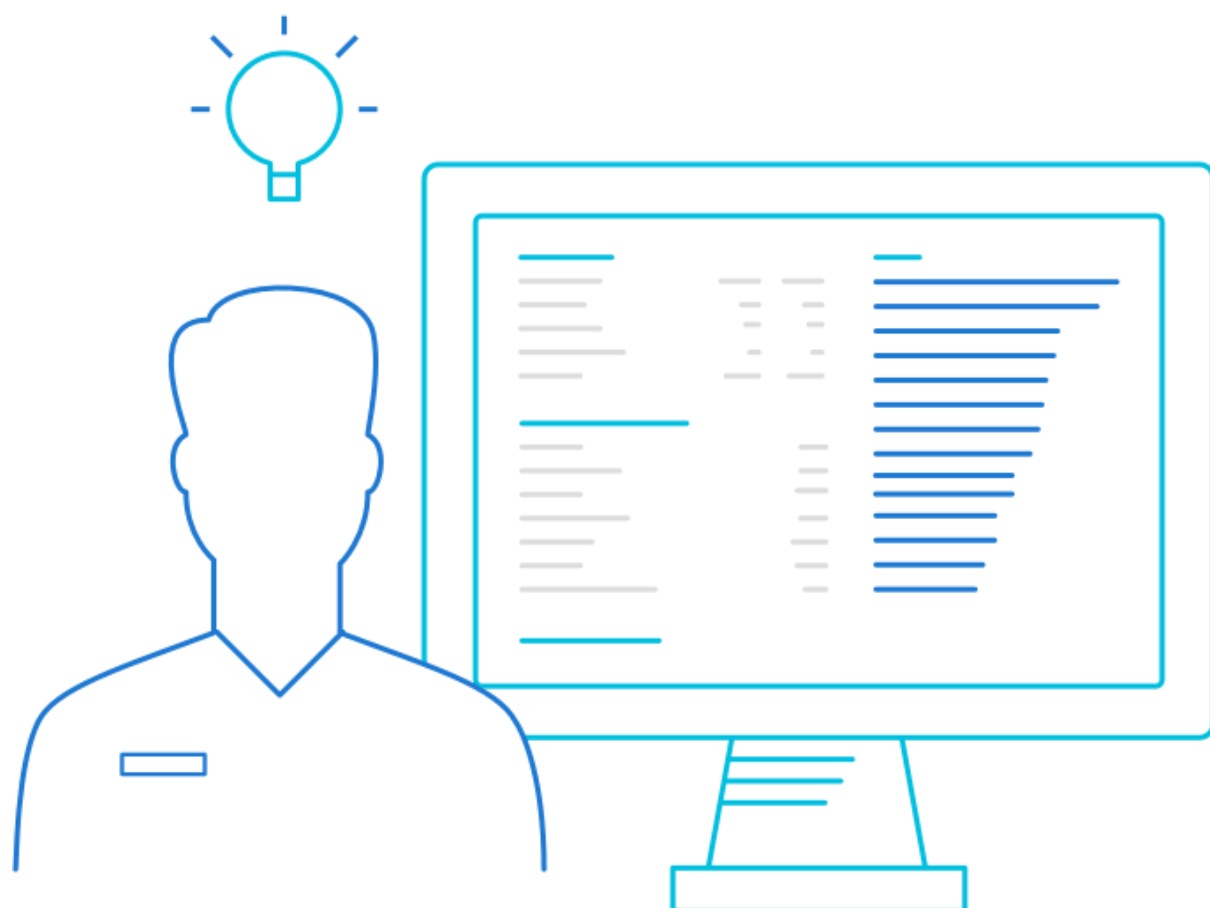


Data Cleansing made simple. Quickly and easily
remove data that may distort your analysis.

Trifacta's solution for data cleansing

Trifacta, and a six-step approach to data clean up and data preparation, it's easy to remove or correct records that are inaccurate, missing, or corrupt, whether from a database, a table, or a set of records. This allows organizations to dramatically reduce their time spent on data cleansing, and leads to better, more accurate analysis. Trifacta helps businesses reduce the amount of resources spent on data scrubbing. With an easier data scrubbing process, businesses are able to analyze more data and get more out of the analysis.

To learn more about Trifacta's unique approach to data cleaning, data scrubbing and data wrangling, check out our eBook, *Six Core Wrangling Activities*.



Boost results and revenue

Clean data makes for better results and greater ROI on marketing and communications campaigns.

Delivering a targeted and consistent message to appropriate audiences will positively impact results and generate far higher response rates for campaigns.

This increase in responses and interactions will help achieve overall business goals and drive revenue.

Clean data also helps marketers to identify high-value prospects more easily. By segmenting data sets marketers can target individuals with personalised messages that are more likely to generate high-value business.

Save money and reduce waste

Data cleansing reduces the waste associated with physical marketing strategies like direct mail marketing.

With an up-to-date data list, you can ensure you are contacting people that have a genuine interest in your message. This greatly reduces the likelihood of your mailing being thrown away before it's read.

Cleansing also helps by removing incorrect details that may affect mailing accuracy. This includes details on people that have changed work/home address or even passed away.

By excluding these contacts you reduce the amount of printing and distribution required for mailings. This saves money and minimises the environmental damage of your campaigns.

Save time and increase productivity

How many hours per month are wasted by sales and marketing teams on calls and emails to expired contacts or people who simply aren't interested?

Probably more than you would like to admit.

More accurate data reduces the time wasted contacting invalid prospects and customers by phone or email. By maintaining a quality data set you can boost the productivity of staff and positively impact the business as a whole.

Additional Benefits of Data Cleansing

Protect reputation

People don't want to receive information that has no relevance to them. More often than not spam mail and unsolicited contact will damage the reputation of your brand.

By maintaining accurate data you ensure that your communications only reach people that would benefit from them. Not only does this increase the likelihood of generating business, it also helps maintain your brand integrity and reputation.

Minimise compliance risks

The introduction of the GDPR, data security and permissions are more important than ever before.

Regular cleansing of databases helps businesses to keep tabs on customer contact permissions and ensure people who have opted out of communications are not contacted by the organisation.

Maintaining a clean and organised data set will help avoid the hefty fines associated with breaching GDPR and other legislation.

PUBLICATIONS (Optional) REFERENCES

7 REFERENCES

1. [1] J. F. Islam, M. Mondal, and C. K. Roy, “Bug Replication in Code Clones: An Empirical Study,” in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 68–78.
2. [2] C. K. Roy, M. F. Zibran, and R. Koschke, “The vision of software clone management: Past, present, and future (Keynote paper),” in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 18–33.