Unit II: Data Cleaning

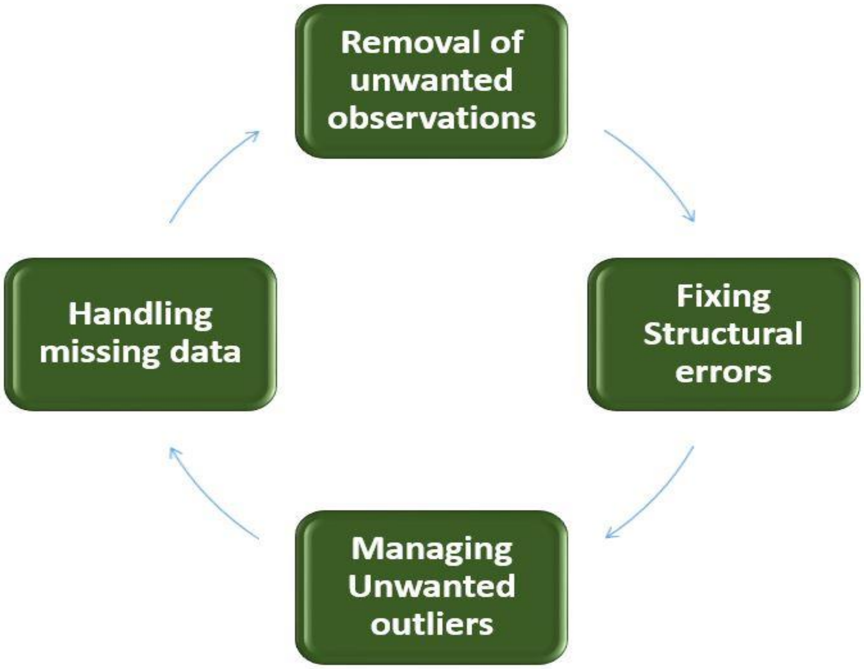
Data cleaning process in research
Data cleansing or **data cleaning** is the **process** of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or **database** and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the **data** and then replacing, modifying, or deleting the dirty or coarse **data**.

ntroduction:

Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. It surely isn't the fanciest part of machine learning and at the same time, there aren't any hidden tricks or secrets to uncover. However, the success or failure of a project relies on proper data cleaning. Professional data scientists usually invest a very large portion of their time in this step because of the belief that **"Better data beats fancier algorithms"**.

If we have a well-cleaned dataset, there are chances that we can get achieve good results with simple algorithms also, which can prove very beneficial at times especially in terms of computation when the dataset size is large.
Obviously, different types of data will require different types of cleaning. However, this systematic approach can always serve as a good starting point.

Steps involved in Data Cleaning:



1. **Removal of unwanted observations**
   This includes deleting duplicate/ redundant or irrelevant values from your dataset. Duplicate observations most frequently arise during data collection and Irrelevant observations are those that don’t actually fit the specific problem that you’re trying to solve.
   - Redundant observations alter the efficiency by a great extent as the data repeats and may add towards the correct side or towards the incorrect side, thereby producing unfaithful results.
   - Irrelevant observations are any type of data that is of no use to us and can be removed directly.
2. **Fixing Structural errors**
   The errors that arise during measurement, transfer of data, or other similar situations are called structural errors. Structural errors include typos in the name of features, the same attribute with a different name, mislabeled classes, i.e. separate classes that should really be the same, or inconsistent capitalization.
   - For example, the model will treat America and America as different classes or values, though they represent the same value or red, yellow, and red-yellow as different classes or attributes, though one class can be included in the other two classes. So, these are some structural errors that make our model inefficient and give poor quality results.
3. **Managing Unwanted outliers**
   Outliers can cause problems with certain types of models. For example, linear regression models are less robust to outliers than decision tree models. Generally, we should not remove outliers until we have a legitimate reason to remove them. Sometimes, removing them improves performance, sometimes not. So, one must have a good reason to remove the outlier, such as suspicious measurements that are unlikely to be part of real data.
4. **Handling missing data**
   Missing data is a deceptively tricky issue in machine learning. We cannot just ignore or remove the missing observation. They must be handled carefully as they can be an indication of something important. The two most common ways to deal with missing data are:
   - Dropping observations with missing values.
     - The fact that the value was missing may be informative in itself.
     - Plus, in the real world, you often need to make predictions on new data even if some of the features are missing!
   - Imputing the missing values from past observations.
     - Again, “missingness” is almost always informative in itself, and you should tell your algorithm if a value was missing.
     - Even if you build a model to impute your values, you’re not adding any real information. You’re just reinforcing the patterns already provided by other features.

Missing data is like missing a puzzle piece. If you drop it, that's like pretending the puzzle slot isn't there. If you impute it, that's like trying to squeeze in a piece from somewhere else in the puzzle.
So, missing data is always an informative and an indication of something important. And we must be aware of our algorithm of missing data by flagging it. By using this technique of flagging and filling, you are essentially allowing the algorithm to estimate the optimal constant for missingness, instead of just filling it in with the mean.

What are the steps in data cleaning?
Data cleaning in six steps

1. Monitor errors. Keep a record of trends where most of your errors are coming from. ...
2. Standardize your process. Standardize the point of entry to help reduce the risk of duplication.
3. Validate **data** accuracy. ...
4. Scrub for duplicate **data**. ...
5. Analyze your **data**. ...
6. Communicate with your team.

How do you clean inconsistent data?
There are 3 main approaches to cleaning missing data:

1. Drop rows and/or columns with missing **data**. ...
2. Recode missing **data** into a different format. ...
3. Fill in missing values with "best guesses." Use moving averages and backfilling to estimate the most probable values of **data** at that point.

What are the best practices for data cleaning?
5 Best Practices for Data Cleaning

1. Develop a Data Quality Plan. Set expectations. ...
2. **Standardize** Contact Data at the Point of Entry. The entry of data is the first cause of dirty data. ...
3. Validate the Accuracy of Your Data. So how can you validate the accuracy of your data in real time? ...
4. Identify Duplicates. ...
5. Append Data.

   What is data cleaning in data analysis?
6. **Data cleaning** is the process of preparing **data** for **analysis** by removing or modifying **data** that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. This **data** is usually not necessary or helpful when it comes to analyzing **data** because it may hinder the process or provide inaccurate results.

What are examples of dirty data?

The 7 Types of Dirty Data

- Duplicate **Data**.
- Outdated **Data**.
- Insecure **Data**.
- Incomplete **Data**.
- Incorrect/Inaccurate **Data**.
- Inconsistent **Data**.
- Too Much **Data**.

What are data cleansing tools?

Download our Mobile App

- OpenRefine: Formerly known as Google Refine, this powerful tool comes handy for dealing with messy **data**, **cleaning** and transforming it. ...
- Trifacta Wrangler: ...
- Drake: ...
- TIBCO Clarity: ...
- Winpure: ...
- **Data** Ladder: ...
- **Data** Cleaner:

2 Why Data Cleaning is Necessary

Data cleaning might seem dull and uninteresting, but it’s one of the most important tasks you would have to do as a data science professional. Having wrong or bad quality data can be detrimental to your processes and analysis. Poor data can cause a stellar algorithm to fail.

On the other hand, high-quality data can cause a simple algorithm to give you outstanding results. There are many data cleaning techniques, and you should get familiar with them to improve your data quality. Not all data is useful. So that’s another major factor which affects your data quality.

There are many reasons why data cleaning is essential. Some of them are listed below:

Efficiency

Having clean data (free from wrong and inconsistent values) can help you in performing your analysis a lot faster. You’d save a considerable amount of time by doing this task beforehand. When you clean your data before using it, you’d be able to avoid multiple errors. If you use data containing false values, your results won’t be accurate.

And the chances are, you would have to redo the entire task again, which can cause a lot of waste of time. If you choose to clean your data before using it, you can generate results faster and avoid redoing the entire task again.

Error Margin

When you don’t use accurate data for analysis, you will surely make mistakes. Suppose, you’ve gotten a lot of effort and time into analyzing a specific group of datasets. You are very eager to show the results to your superior, but in the meeting, your superior points out a few mistakes the situation gets kind of embarrassing and painful.

Wouldn’t you want to avoid such mistakes from happening? Not only do they cause embarrassment, but they also waste resources. Data cleansing helps you in that regard full stop it is a widespread practice, and you should learn the methods used to clean data.

Using a simple algorithm with clean data is way better than using an advanced with unclean data.

Determining Data Quality

Is The Data Valid? (Validity)

The validity of your data is the degree to which it follows the rules of your particular requirements. For example, you how to import phone numbers of different customers, but in some places, you added email addresses in the data. Now because your needs were explicitly for phone numbers, the email addresses would be invalid.

Validity errors take place when the input method isn’t properly inspected. You might be using spreadsheets for collecting your data. And you might enter the wrong information in the cells of the spreadsheet.

There are **multiple kinds of constraints** your data has to conform to for being valid. Here they are:

Range:

Some types of numbers have to be in a specific range. For example, the number of products you can transport in a day must have a minimum and maximum value. There would surely be a particular range for the data. There would be a starting point and an end-point.

Data-Type:

Some data cells might require a specific kind of data, such as numeric, Boolean, etc. For example, in a Boolean section, you wouldn’t add a numerical value.

Compulsory constraints:

In every scenario, there are some mandatory constraints your data should follow. The compulsory restrictions depend on your specific needs. Surely, specific columns of your data shouldn't be empty.For example, in the list of your clients' names, the column of 'name' can't be empty.

Cross-field examination:

There are certain conditions which affect multiple fields of data in a particular form. Suppose the time of departure of a flight couldn't be earlier than it's arrival. In a balance sheet, the sum of the debit and credit of the client must be the same. It can't be different.

These values are related to each other, and that's why you might need to perform cross-field examination.

Unique Requirements:

Particulars types of data have unique restrictions. Two customers can't have the same customer support ticket. Such kind of data must be unique to a particular field and can't be shared by multiple ones.

Set-Membership Restrictions:

Some values are restricted to a particular set. Like, gender can either be Male, Female or Unknown.

Regular Patterns:

Some pieces of data follow a specific format. For example, email addresses have the format 'randomperson@randomemail.com'. Similarly, phone numbers have ten digits.

If the data isn't in the required format, it would also be invalid.

If a person omits the '@' while entering an email address, then the email address would be invalid, wouldn't it? Checking the validity of your data is the first step to determine its quality. Most of the time, the cause of entry of invalid information is human error.

Getting rid of it will help you in streamlining your process and avoiding useless data values beforehand.

Accuracy

Now that you know that most of the data you have is valid, you'll have to focus on establishing its accuracy. Even though the data is valid, it doesn't mean the data is accurate. And determining accuracy helps you to figure out if the data you entered was accurate or not.

The address of a client could be in the right format, but it doesn't need to be the right one. Maybe the email has an additional digit or character that makes it wrong. Another example is of the phone number of a customer.

If the phone number has all the digits, it's a valid value. But that doesn't mean it's true. When you have definitions for valid values, figuring out the invalid ones is easy. But that doesn't help with checking the accuracy of the same. Checking the accuracy of your data values requires you to use third-party sources.

This means you'll have to rely on data sources different from the one you're using currently. You'll have to cross-check your data to figure out if it's accurate or not. Data cleaning techniques don't have many solutions for checking the accuracy of data values.

However, depending on the kind of data you're using, you might be able to find resources that could help you in this regard. You shouldn't confuse accuracy with precision.

Accuracy vs Precision

While accuracy relies on establishing whether your entered data was correct or not, precision requires you to give more details about the same. A customer might enter a first name in your data field. But if there's no last name, it'd be challenging to be more precise.

Another example can be of an address. Suppose you ask a person where he/she lives. They might say that they live in London. That could be true. However, that's not a precise answer because you don't know where they live in London.

A precise answer would be to give you a street address.

Completeness

It's nearly impossible to have all the info you need. Completeness is the degree to which you know all the required values. Completeness is a little more challenging to achieve than accuracy or validity. That's because you can't assume a value. You only have to enter known facts.

You can try to complete your data by redoing the data gathering activities (approaching the clients again, re-interviewing people, etc.). But that doesn't mean you'd be able to complete your data thoroughly.

Suppose you re-interview people for the data you needed earlier. Now, this scenario has the problem of recall. If you ask them the same questions again, chances are, they might not remember what they had answered before. This can lead to them, giving you the wrong answer.

You might ask him what books they were reading five months ago. And they might not remember. Similarly, you might need to enter every customer's contact information. But some of them may not have email addresses. In this case, you'd have to leave those columns empty.

If you have a system which requires you to fill all columns, you can try to enter ‘missing’ or ‘unknown’ there. But entering such values doesn’t mean the data is complete. It would be still be referred to as incomplete.

Consistency

Next to completeness comes consistency. You can measure consistency by comparing two similar systems. Or, you can check the data values within the same dataset to see if they are consistent or not. Consistency can be relational. For example, a customer’s age might be 15, which is a valid value and could be accurate, but they might also be stated senior-citizen in the same system.

In such cases, you’ll need to cross-check the data, similar to measuring accuracy, and see which value is true. Is the client a 15-year old? Or is the client a senior-citizen? Only one of these values could be true.

There are multiple ways to make your data consistent.

Check different systems:

You can take a look at another similar system to find whether the value you have is real or not. If two of your systems are contradicting each other, it might help to check the third one.

In our previous example, suppose you check the third system and find the age of the customer is 65. This shows that the second system, which said the customer is a senior citizen, would hold.

Check the latest data:

Another way to improve the consistency of your data is to check the more recent value. It can be more beneficial to you in specific scenarios. You might have two different contact numbers for a customer in your record. The most recent one would probably be more reliable because it’s possible that the customer switched numbers.

Check the source:

The most fool-proof way to check the reliability of the data is to contact the source simply. In our example of the customer’s age, you can opt to contact the customer directly and ask them their age. However, it’s not possible in every scenario and directly contacting the source can be highly tricky. Maybe the customer doesn’t respond, or their contact information isn’t available.

Uniformity

You should ensure that all the values you’ve entered in your dataset are in the same units. If you’re entering SI units for measurements, you can’t use the Imperial system in some places. On the other hand, if at one place you’ve entered the time in seconds, then you should enter it in this format all across the dataset.

Checking the uniformity of your records is quite easy. A simple inspection can reveal whether a particular value is in the required unit or not. The units you use for entering your data depend on your specific requirements.

Data Cleansing Techniques

Your choice of data cleaning techniques relies on a lot of factors. First, what kind of data are you dealing with? Are they numeric values or strings? Unless you have too few values to handle, you shouldn’t expect to clean your data with just one technique as well.

You might need to use multiple techniques for a better result. The more data types you have to handle, the more cleansing techniques you’ll have to use. Being familiar with all of these methods will help you in rectifying errors and getting rid of useless data.

1. Remove Irrelevant Values

The first and foremost thing you should do is remove useless pieces of data from your system. Any useless or irrelevant data is the one you don’t need. It might not fit the context of your issue.

You might only have to measure the average age of your sales staff. Then their email address wouldn’t be required. Another example is you might be checking to see how many customers you contacted in a month. In this case, you wouldn’t need the data of people you reached in a prior month.

However, before you remove a particular piece of data, make sure that it is irrelevant because you might need it to check its correlated values later on (for checking the consistency). And if you can get a second opinion from a more experienced expert before removing data, feel free to do so.

You wouldn’t want to delete some values and regret the decision later on. But once you’re assured that the data is irrelevant, get rid of it.

2. Get Rid of Duplicate Values

Duplicates are similar to useless values – You don’t need them. They only increase the amount of data you have and waste your time. You can get rid of them with simple searches. Duplicate values could be present in your system for several reasons.

Maybe you combined the data of multiple sources. Or, perhaps the person submitting the data repeated a value mistakingly. Some user clicked twice on ‘enter’ when they were filling an online form. You should remove the duplicates as soon as you find them.

3. Avoid Typos (and similar errors)

Typos are a result of human error and can be present anywhere. You can fix typos through multiple algorithms and techniques. You can map the values and convert them into the correct spelling. Typos are essential to fix because models treat different values differently. Strings rely a lot on their spellings and cases.

‘George’ is different from ‘george’ even though they have the same spelling. Similarly ‘Mike’ and ‘Mice’ are different from each other, also though they have the same number of characters. You’ll need to look for typos such as this and fix them appropriately.

Another error similar to typos is of strings’ size. You might need to pad them to keep them in the same format. For example, your dataset might require you to have 5-digit numbers only. So if you have any value which only has four digits such as ‘3994’ you can add a zero in the beginning to increase its number of digits.

Its value would remain the same as ‘03994’, but it’ll keep your data uniform. An additional error with strings is of white spaces. Make sure you remove them from your strings to keep them consistent.

4. Convert Data Types

Data types should be uniform across your dataset. A string can’t be numeric nor can a numeric be a boolean. There are several things you should keep in mind when it comes to converting data types:

- Keep numeric values as numerics
- Check whether a numeric is a string or not. If you entered it as a string, it would be incorrect.
- If you can’t convert a specific data value, you should enter ‘NA value’ or something of this sort. Make sure you add a warning as well to show that this particular value is wrong.

5. Take Care of Missing Values

There would always be a piece of missing data. You can’t avoid it. So you should know how to handle them to keep your data clean and free from errors. A particular column in your dataset may have too many missing values. In that case, it would be wise to get rid of the entire column because it doesn’t have enough data to work with.

Point to note: You shouldn’t ignore missing values.

Ignoring missing values can be a significant mistake because they will contaminate your data, and you won’t get accurate results. There are multiple ways to deal with missing values.

Imputing Missing Values:

You can impute missing values, which means, assuming the approximate value. You can use linear regression or median to calculate the missing value. However, this method has its implications because you can't be sure if that would be the real value.

Another method to impute missing values is to copy the data from a similar dataset. This method is called 'Hot-deck imputation'. You're adding value in your current record while considering some constraints such as data-type and range.

Highlighting Missing Values:

Imputation isn't always the best measure to take care of missing values. Many experts argue that it only leads to more mixed results as they are not 'real'. So, you can take another approach and inform the model that the data is missing. Telling the model (or the algorithm) that the specific value is unavailable can be a piece of information as well.

If random reasons aren't responsible for your missing values, it can be beneficial to highlight or flag them. For example, your records may not have many answers to a specific question of your survey because your customer didn't want to answer it in the first place.

If the missing value is numeric, you can use 0. Just make sure that you ignore these values during statistical analysis. On the other hand, if the missing value is a categorical value, you can fill 'missing'.

What is data cleaning?

Data cleaning is the process of ensuring data is correct, consistent and usable. You can clean data by identifying errors or corruptions, correcting or deleting them, or manually processing data as needed to prevent the same errors from occurring.

Most aspects of data cleaning can be done through the use of software tools, but a portion of it must be done manually. Although this can make data cleaning an overwhelming task, it is an essential part of managing company data.

3 What are the benefits of data cleaning?

There are many benefits to having clean data:

1. It removes major errors and inconsistencies that are inevitable when multiple sources of data are being pulled into one dataset.
2. Using tools to clean up data will make everyone on your team more efficient as you'll be able to quickly get what you need from the data available to you.
3. Fewer errors means happier customers and fewer frustrated employees.
4. It allows you to map different data functions, and better understand what your data is intended to do, and learn where it is coming from.

3 Data cleaning in six steps

The first step before starting a data cleaning project is to first look at the big picture. Ask yourself: What are your goals and expectations?

To achieve those goals you’ve set, next, you must plan a data cleanup strategy. A great guideline is to focus on your top metrics. Some questions to ask:

- What is your highest metric looking to achieve?
- What is your company’s overall goal and what is each member looking to achieve from it?

A good way to start is to get the key stakeholders together and brainstorm.

Here are some best practices when it comes to create a data cleaning process:

1. Monitor errors

Keep a record of trends where most of your errors are coming from.This will make it a lot easier to identify and fix incorrect or corrupt data. Records are especially important if you are integrating other solutions with your fleet management software, so that your errors don’t clog up the work of other departments.

2. Standardize your process

Standardize the point of entry to help reduce the risk of duplication.

3. Validate data accuracy

Once you have cleaned your existing database, validate the accuracy of your data. Research and invest in data tools that allow you to clean your data in real-time. Some tools even use AI or **machine learning** to better test for accuracy.

4. Scrub for duplicate data

Identify duplicates to help save time when analyzing data. Repeated data can be avoided by researching and investing in different data cleaning tools that can analyze raw data in bulk and automate the process for you.

5. Analyze your data

After your data has been standardized, validated and scrubbed for duplicates, use third-party sources to append it. Reliable third-party sources can capture information directly from first-party sites, then clean and compile the data to provide more complete information for business intelligence and analytics.

6. Communicate with your team

Share the new standardized cleaning process with your team to promote adoption of the new protocol. Now that you’ve scrubbed down your data, it’s important to keep it clean. Keeping your team in the loop will help you develop and strengthen customer segmentation and send more targeted information to customers and prospects.

Finally, monitor and review data regularly to catch inconsistencies.

Missing Data Imputation Techniques

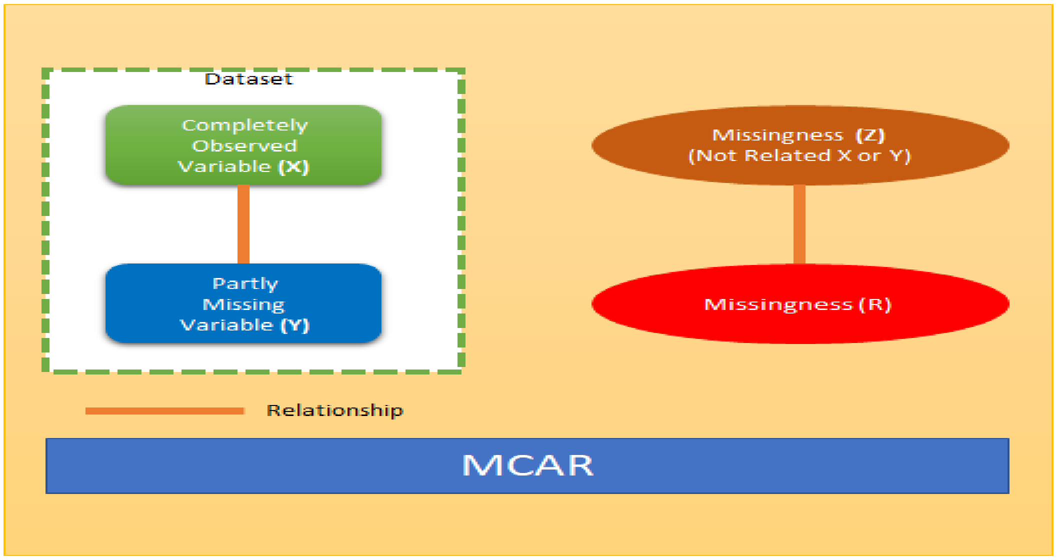
Missing data is an everyday problem that a data professional need to deal with. Though there are many articles, blogs, videos already available, I found it is difficult to find a concise consolidated information in a single place. That's why I am putting my effort here, hoping it will be useful to any data practitioner or enthusiast.

What is missing data? Missing data are defined as values that are not available and that would be meaningful if they are observed. Missing data can be anything from missing sequence, incomplete feature, files missing, information incomplete, data entry error etc. Most datasets in the real world contain missing data. Before you can use data with missing data fields, you need to transform those fields so they can be used for analysis and modelling. Like many other aspects of data science, this too may actually be more art than science. Understanding the data and the domain from which it comes is very important.

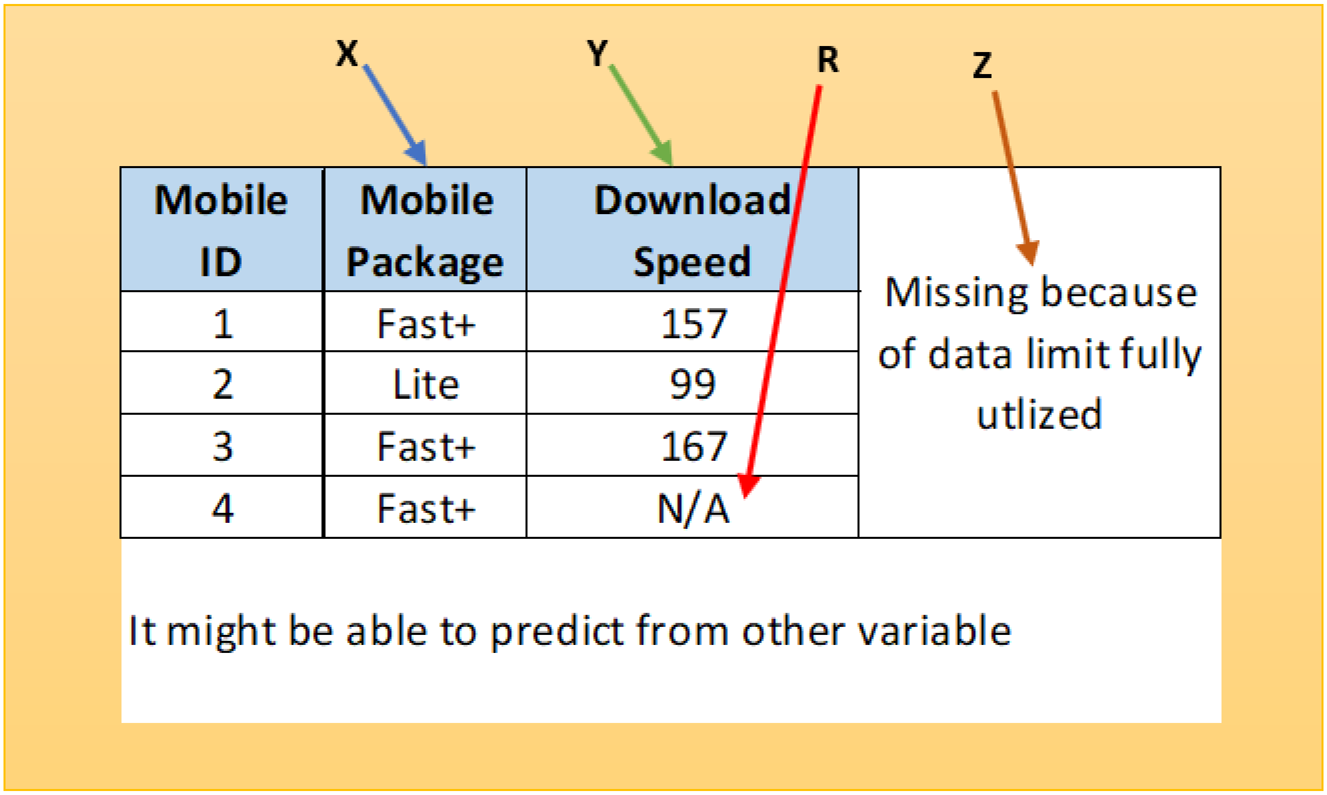
Having missing values in your data is not necessarily a setback but it is an opportunity to perform right feature engineering to guide the model to interpret the missing information right way. There are machine learning algorithms and packages that can automatically detect and deal with missing data, but it's still recommended to transform the missing data manually through analysis and coding strategy. First, we need to understand what are the types of missing data. Missingness is broadly categorized in 3 categories:

Missing Completely at Random (MCAR)

When we say data are **missing completely at random**, we mean that the missingness has nothing to do with the observation being studied (Completely Observed Variable (X) and Partly Missing Variable (Y)). For example, a weighing scale that ran out of batteries, a questionnaire might be lost in the post, or a blood sample might be damaged in the lab. MCAR is an ideal but unreasonable assumption. Generally, data are regarded as being MCAR when data are missing by design, because of an equipment failure or because the samples are lost in transit or technically unsatisfactory. The statistical advantage of data that are MCAR is that the analysis remains **unbiased**. A pictorial view of MCAR as below where missingness has **no relation to dataset variables X or Y**. Missingness is not related to X or Y but some other reason Z.


Dataset
Completely Observed Variable (X)
Partly Missing Variable (Y)
Missingness (Z) (Not Related X or Y)
Missingness (R)
Relationship
MCAR

Let’s explore one example of mobile data. Here one sample has missing value which is not because of dataset variables but because of another reason.



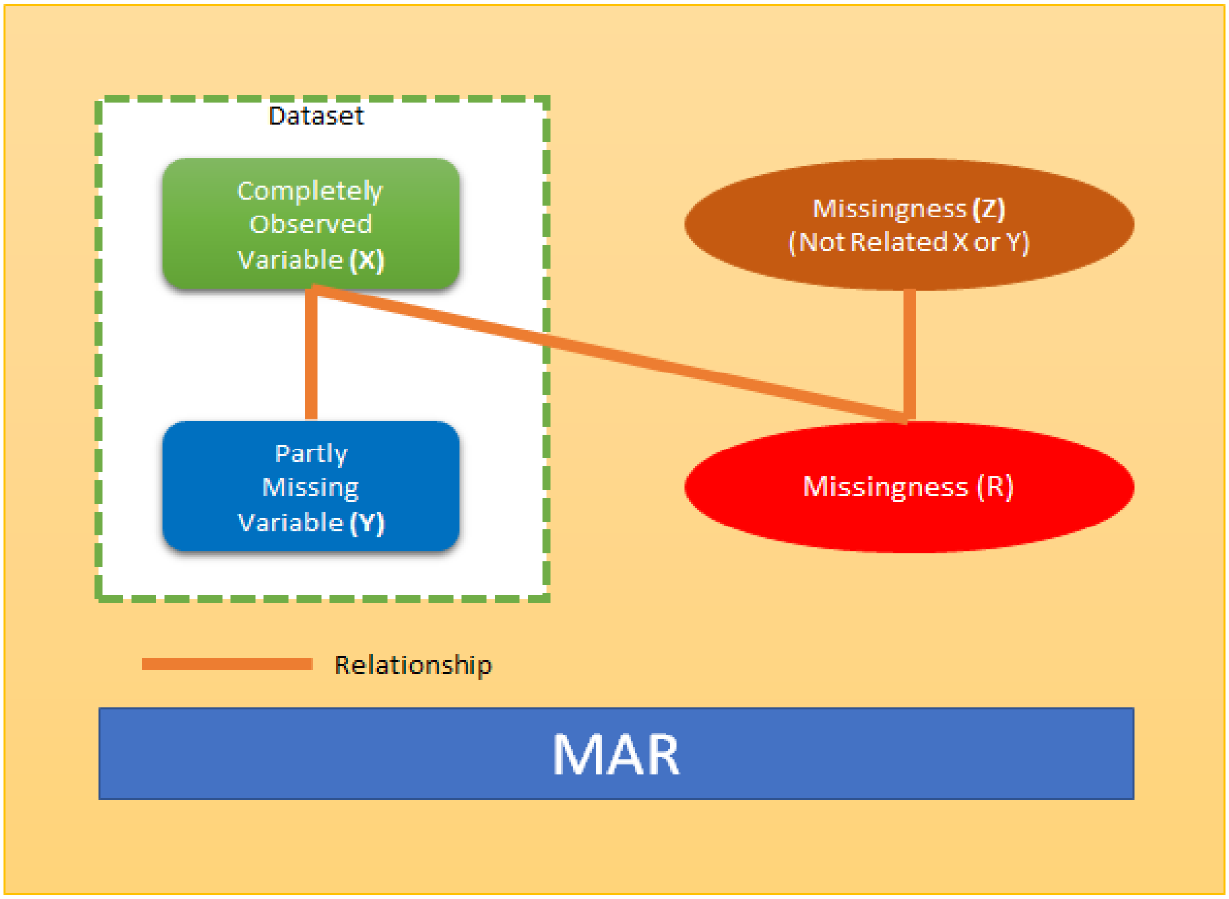
Mobile ID	Mobile Package	Download Speed	
1	Fast+	157	Missing because of data limit fully utlized
2	Lite	99	
3	Fast+	167	
4	Fast+	N/A	

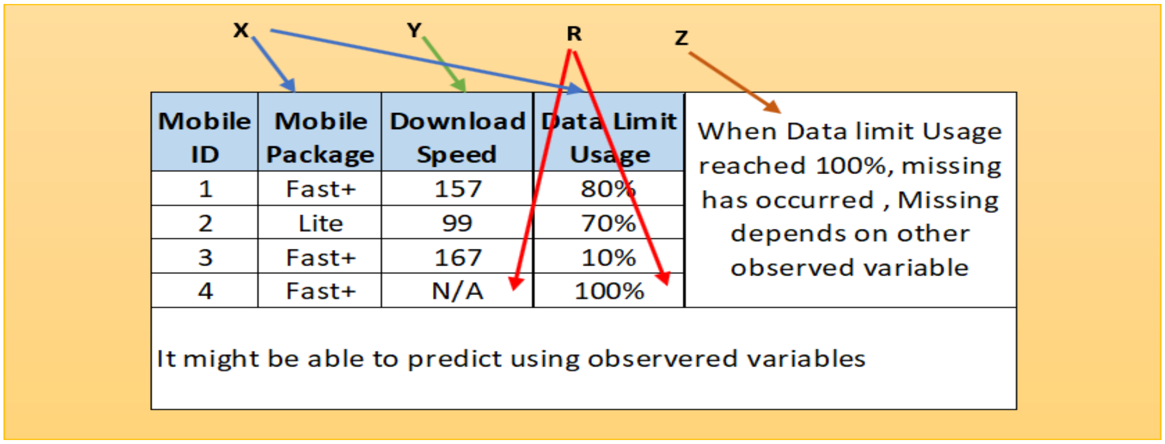
It might be able to predict from other variable

Missing at Random (MAR)

When we say data are **missing at random**, we mean that missing data on a partly missing variable (Y) is related to some other completely observed variables(X) in the analysis model but not to the values of Y itself.

It is not specifically related to the missing information. For example, if a child does not attend an examination because the child is ill, this might be predictable from other data we have about the child's health, but it would not be related to what we would have examined had the child not been ill. Some may think that MAR does not present a problem. However, MAR does not mean that the missing data can be ignored. A pictorial view of MAR as below where missingness has relation to **dataset variable X but not with Y**. It can have other relation as well (Z).


Dataset
Completely Observed Variable (X)
Partly Missing Variable (Y)
Missingness (Z) (Not Related X or Y)
Missingness (R)
Relationship
MAR



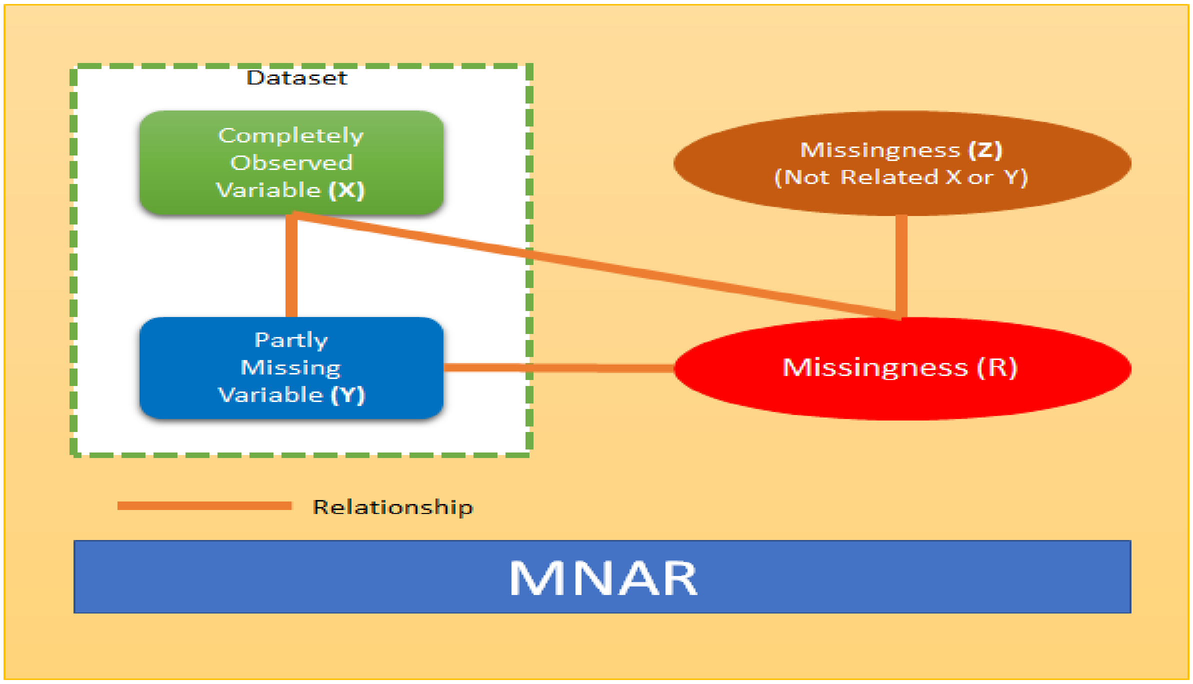
Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	100%

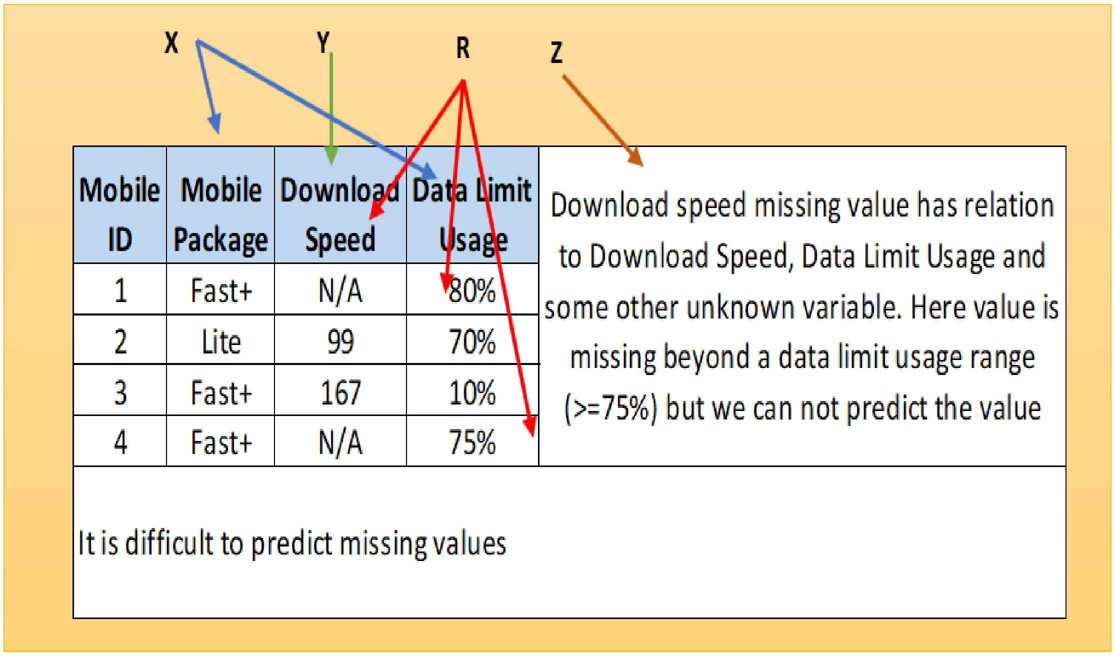
When Data limit Usage reached 100%, missing has occurred , Missing depends on other observed variable

It might be able to predict using observered variables

Missing not at Random (MNAR)

If the characters of the data do not meet those of MCAR or MAR, then they fall into the category of missing not at random (MNAR). When data are **missing not at random**, the missingness is specifically related to what is missing, e.g. a person does not attend a drug test because the person took drugs the night before, a person did not take English proficiency test due to his poor English language skill. The cases of MNAR data are problematic. The only way to obtain an unbiased estimate of the parameters in such a case is to model the missing data but that requires proper understanding and domain knowledge of the missing variable. The model may then be incorporated into a more complex one for estimating the missing values. A pictorial view of MNAR as below where missingness has **direct relation to variable Y**. It can have other relation as well (X & Z).


Dataset
Completely Observed Variable **(X)**
Partly Missing Variable **(Y)**
Missingness **(Z)** (Not Related X or Y)
Missingness (R)
Relationship
MNAR



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	N/A	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	75%

Download speed missing value has relation to Download Speed, Data Limit Usage and some other unknown variable. Here value is missing beyond a data limit usage range (>=75%) but we can not predict the value

It is difficult to predict missing values

There are several strategies which can be applied to handle missing data to make the Machine Learning/Statistical Model.

Try to obtain the missing data

This may be possible in data collection phase in survey like situation where one can check if survey data is captured in entirety before respondent leaves the room. Sometimes it may be possible to reach out to the source to get the data like asking the missing question again for a response. In real world scenario, this is very unlikely way to resolve the missing data problem.

Educated Guessing

It sounds arbitrary and isn’t never preferred course of action, but one can sometimes infer a missing value based on other response. For related questions, for example, like those often presented in a matrix, if the participant responds with all “2s”, assume that the missing value is a 2.

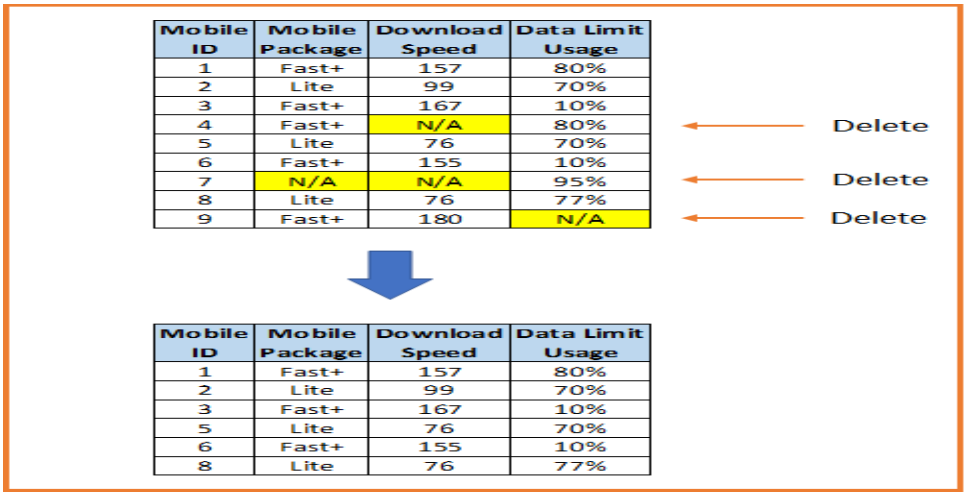
Discard Data

1) list-wise (Complete-case analysis — CCA) deletion

By far the most common approach to the missing data is to simply omit those cases with the missing data and analyse the remaining data. This approach is known as the complete case (or available case) analysis or list-wise deletion.

If there is a large enough sample, where power is not an issue, and the assumption of MCAR is satisfied, the listwise deletion may be a reasonable strategy. However, when there is not a large sample, or the assumption of MCAR is not satisfied, the listwise deletion is not the optimal strategy. It also introduces bias if it does not satisfy MCAR.

Refer to below sample observation after deletion



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	80%
5	Lite	76	70%
6	Fast+	155	10%
7	N/A	N/A	95%
8	Lite	76	77%
9	Fast+	180	N/A

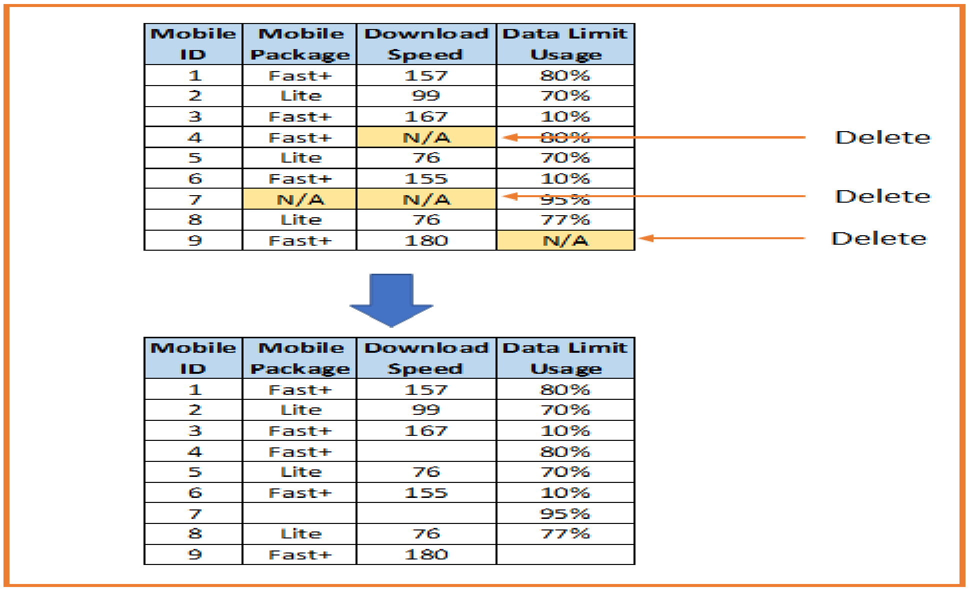
Delete
Delete
Delete

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
5	Lite	76	70%
6	Fast+	155	10%
8	Lite	76	77%

2) Pairwise (available case analysis — ACA) Deletion

In this case, only the missing observations are ignored and analysis is done on variables present. If there is missing data elsewhere in the data set, the existing values are used. Since a pairwise deletion uses all information observed, it preserves more information than the listwise deletion.

Pairwise deletion is known to be less biased for the MCAR or MAR data. However, if there are many missing observations, the analysis will be deficient. the problem with pairwise deletion is that even though it takes the available cases, one can’t compare analyses because the sample is different every time.



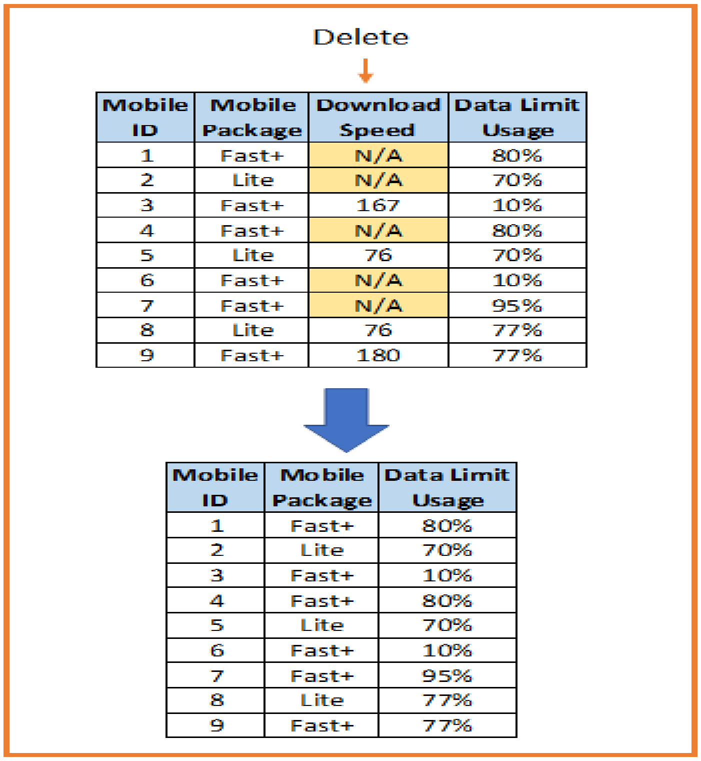
Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	80%
5	Lite	76	70%
6	Fast+	155	10%
7	N/A	N/A	95%
8	Lite	76	77%
9	Fast+	180	N/A

Delete
Delete
Delete

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+		80%
5	Lite	76	70%
6	Fast+	155	10%
7			95%
8	Lite	76	77%
9	Fast+	180	

3) Dropping Variables

If there are too many data missing for a variable it may be an option to delete the variable or the column from the dataset. There is no rule of thumbs for this but depends on situation and a proper analysis of data is needed before the variable is dropped all together. This should be the last option and need to check if model performance improves after deletion of variable.



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	N/A	80%
2	Lite	N/A	70%
3	Fast+	167	10%
4	Fast+	N/A	80%
5	Lite	76	70%
6	Fast+	N/A	10%
7	Fast+	N/A	95%
8	Lite	76	77%
9	Fast+	180	77%

Mobile ID	Mobile Package	Data Limit Usage
1	Fast+	80%
2	Lite	70%
3	Fast+	10%
4	Fast+	80%
5	Lite	70%
6	Fast+	10%
7	Fast+	95%
8	Lite	77%
9	Fast+	77%

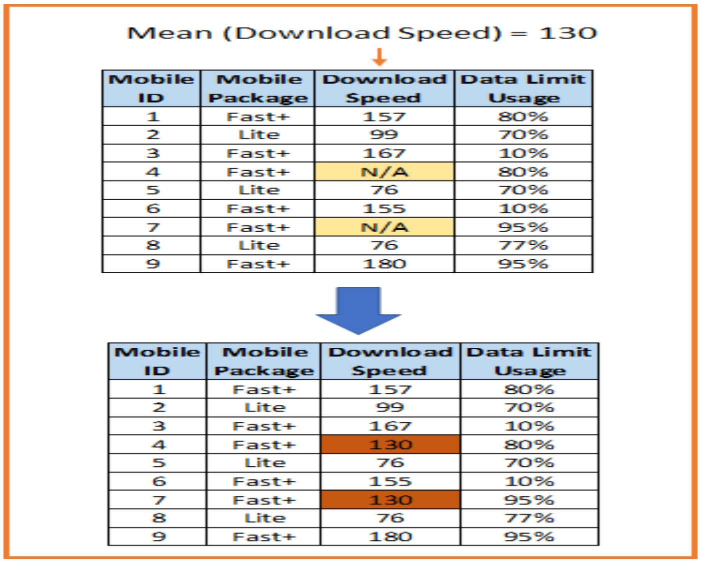
Retain All Data

The goal of any imputation technique is to produce a complete dataset that can then be then used for machine learning. There are few ways we can do imputation to retain all data for analysis and building the model.

1) Mean, Median and Mode

In this imputation technique goal is to replace missing data with statistical estimates of the missing values. Mean, Median or Mode can be used as imputation value.

In a mean substitution, the mean value of a variable is used in place of the missing data value for that same variable. This has the benefit of not changing the sample mean for that variable. The theoretical background of the mean substitution is that the mean is a reasonable estimate for a randomly selected observation from a normal distribution. However, with missing values that are not strictly random, especially in the presence of a great inequality in the number of missing values for the different variables, the mean substitution method may lead to inconsistent bias. Distortion of original variance and Distortion of co-variance with remaining variables within the dataset are two major drawbacks of this method.

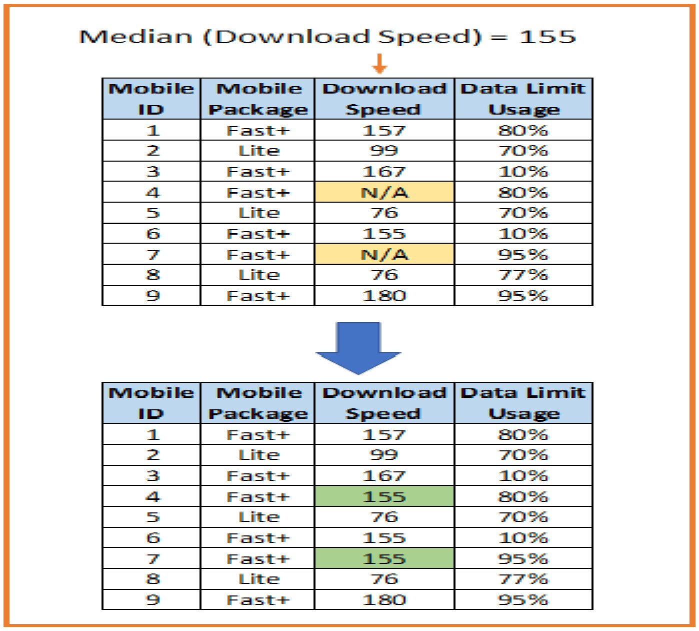


Mean (Download Speed) = 130

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	N/A	95%
8	Lite	76	77%
9	Fast+	180	95%

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	130	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	130	95%
8	Lite	76	77%
9	Fast+	180	95%

Median can be used when variable has a skewed distribution.

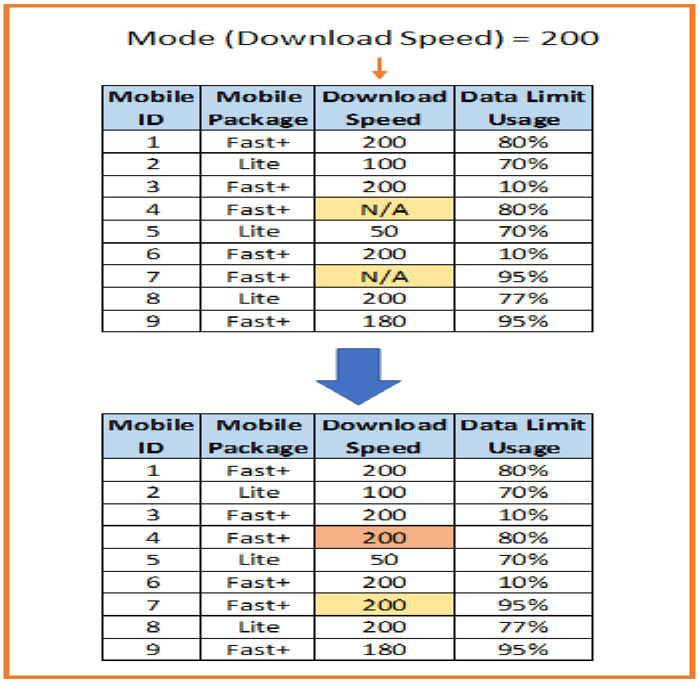


Median (Download Speed) = 155

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	N/A	95%
8	Lite	76	77%
9	Fast+	180	95%

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	155	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	155	95%
8	Lite	76	77%
9	Fast+	180	95%

The rationale for Mode is to replace the population of missing values with the most frequent value, since this is the most likely occurrence.



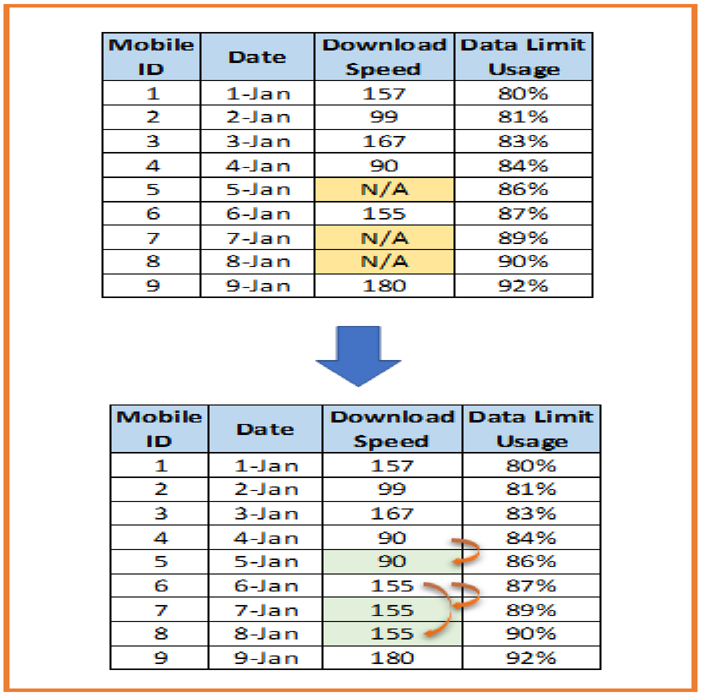
Mode (Download Speed) = 200

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	200	80%
2	Lite	100	70%
3	Fast+	200	10%
4	Fast+	N/A	80%
5	Lite	50	70%
6	Fast+	200	10%
7	Fast+	N/A	95%
8	Lite	200	77%
9	Fast+	180	95%

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	200	80%
2	Lite	100	70%
3	Fast+	200	10%
4	Fast+	200	80%
5	Lite	50	70%
6	Fast+	200	10%
7	Fast+	200	95%
8	Lite	200	77%
9	Fast+	180	95%

2) Last Observation Carried Forward (LOCF)

If data is time-series data, one of the most widely used imputation methods is the last observation carried forward (LOCF). Whenever a value is missing, it is replaced with the last observed value. This method is advantageous as it is easy to understand and communicate. Although simple, this method strongly assumes that the value of the outcome remains unchanged by the missing data, which seems unlikely in many settings.

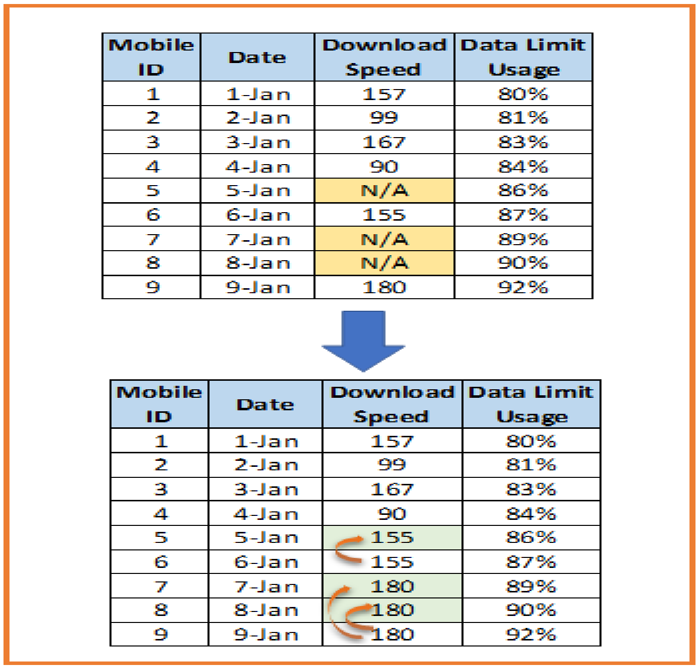


Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	90	86%
6	6-Jan	155	87%
7	7-Jan	155	89%
8	8-Jan	155	90%
9	9-Jan	180	92%

3) Next Observation Carried Backward (NOCB)

A similar approach like LOCF which works in the opposite direction by taking the first observation after the missing value and **c**arrying it backward ("next observation carried backward", or NOCB).

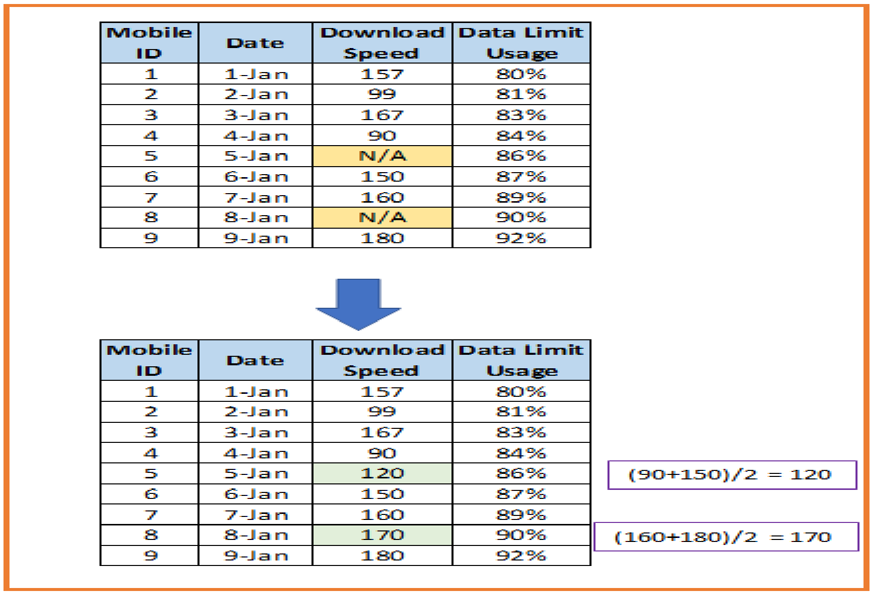


Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	155	86%
6	6-Jan	155	87%
7	7-Jan	180	89%
8	8-Jan	180	90%
9	9-Jan	180	92%

4) Linear Interpolation

Interpolation is a mathematical method that adjusts a function to data and uses this function to extrapolate the missing data. The simplest type of interpolation is the linear interpolation, that makes a mean between the values before the missing data and the value after. Of course, we could have a pretty complex pattern in data and linear interpolation could not be enough. There are several different types of interpolation. Just in Pandas we have the following options like : 'linear', 'time', 'index', 'values', 'nearest', 'zero', 'slinear', 'quadratic', 'cubic', 'polynomial', 'spline', 'piece wise polynomial' and many more .



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	120	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	170	90%
9	9-Jan	180	92%

(90+150)/2 = 120

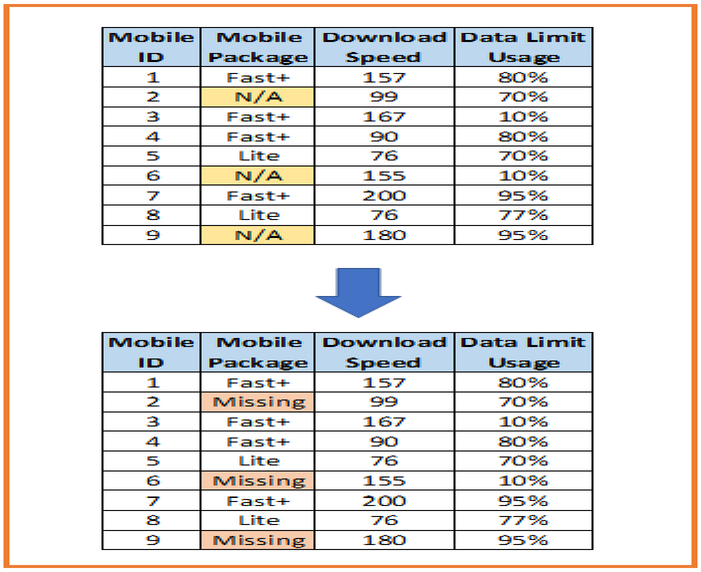
(160+180)/2 = 170

5) Common-Point Imputation

For a rating scale, using the middle point or most commonly chosen value. For example, on a five-point scale, substitute a 3, the midpoint, or a 4, the most common value (in many cases). It is similar to mean value but more suitable for ordinal values.

6) Adding a category to capture NA

This is perhaps the most widely used method of missing data imputation for categorical variables. This method consists in treating missing data as if they were an additional label or category of the variable. All the missing observations are grouped in the newly created label 'Missing'. It does not assume anything on the missingness of the values. It is very well suited when the number of missing data is high.

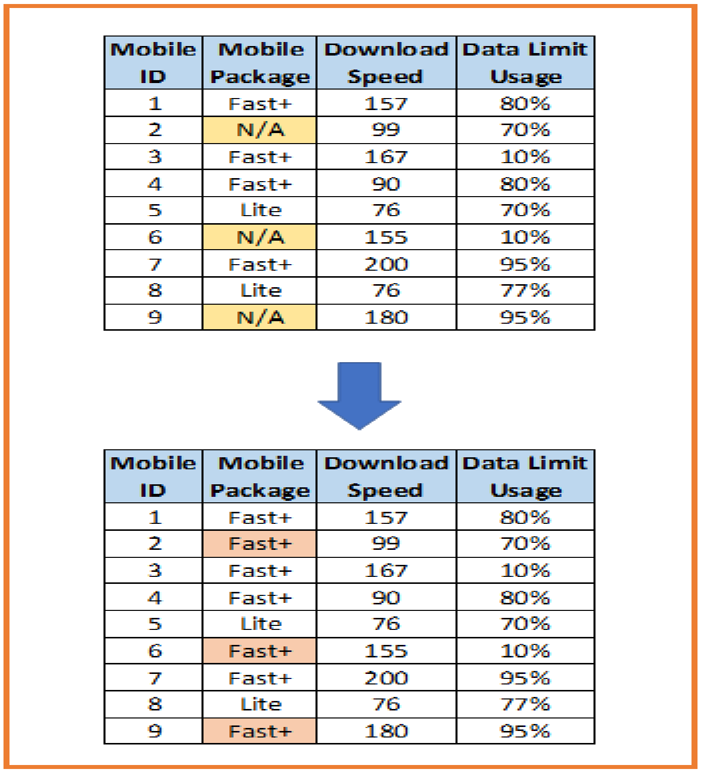


Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	N/A	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	N/A	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	N/A	180	95%

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Missing	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	Missing	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	Missing	180	95%

7) Frequent category imputation

Replacement of missing values by the most frequent category is the equivalent of mean/median imputation. It consists of replacing all occurrences of missing values within a variable by the most frequent label or category of the variable.

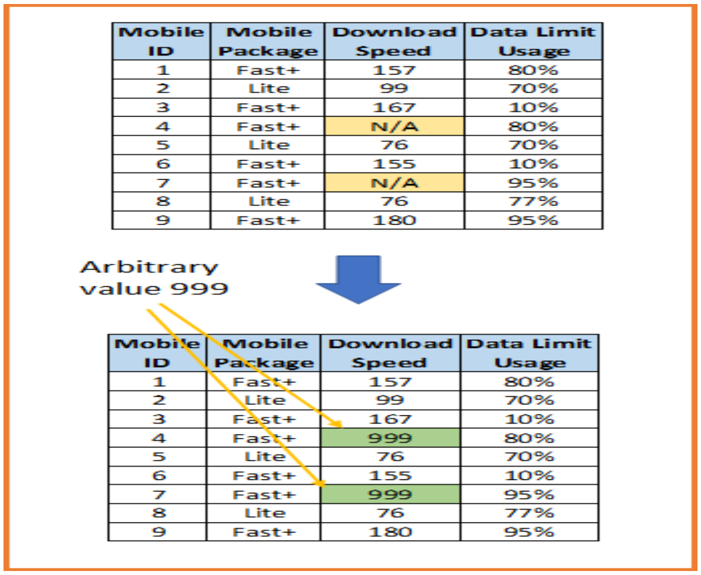


Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	N/A	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	N/A	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	N/A	180	95%

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Fast+	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	Fast+	180	95%

8) Arbitrary Value Imputation

Arbitrary value imputation consists of replacing all occurrences of missing values within a variable by an arbitrary value. Ideally arbitrary value should be different from the median/mean/mode, and not within the normal values of the variable. Typically used arbitrary values are 0, 999, -999 (or other combinations of 9's) or -1 (if the distribution is positive). Sometime data already contain arbitrary value from originator for the missing values. This works reasonably well for numerical features that are predominantly positive in value, and for tree-based models in general. This used to be a more common method in the past when the out-of-the box machine learning libraries and algorithms were not very adept at working with missing data.



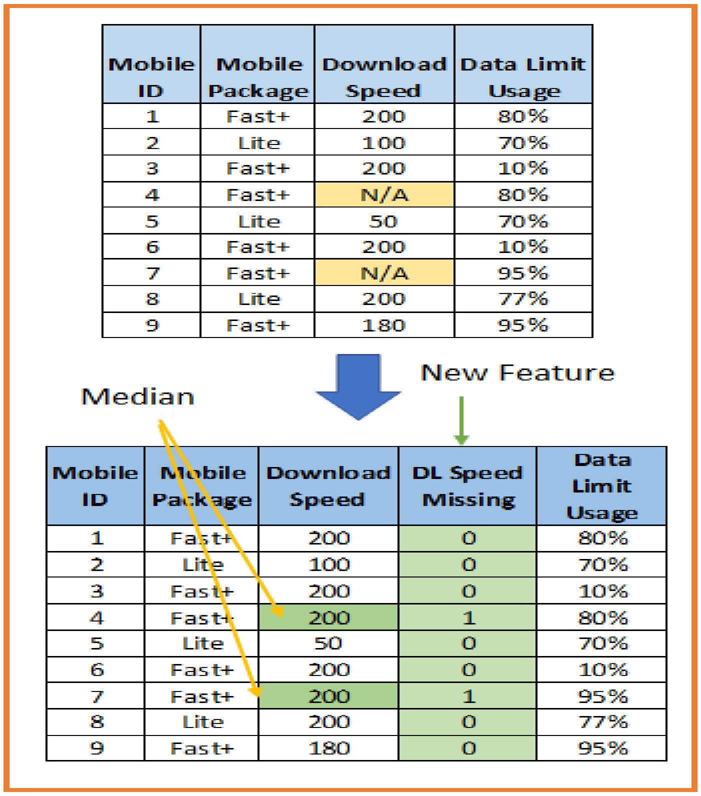
Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	N/A	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	N/A	95%
8	Lite	76	77%
9	Fast+	180	95%

Arbitrary value 999

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Lite	99	70%
3	Fast+	167	10%
4	Fast+	999	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	999	95%
8	Lite	76	77%
9	Fast+	180	95%

9) Adding a variable to capture NA

When data are not missing completely at random, we can capture the importance of missingness by creating an additional variable indicating whether the data was missing for that observation (1) or not (0). The additional variable is a binary variable: it takes only the values 0 and 1, 0 indicating that a value was present for that observation, and 1 indicating that the value was missing for that observation. Typically, mean/median imputation is done together with adding a variable to capture those observations where the data was missing.



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	200	80%
2	Lite	100	70%
3	Fast+	200	10%
4	Fast+	N/A	80%
5	Lite	50	70%
6	Fast+	200	10%
7	Fast+	N/A	95%
8	Lite	200	77%
9	Fast+	180	95%

Median
New Feature

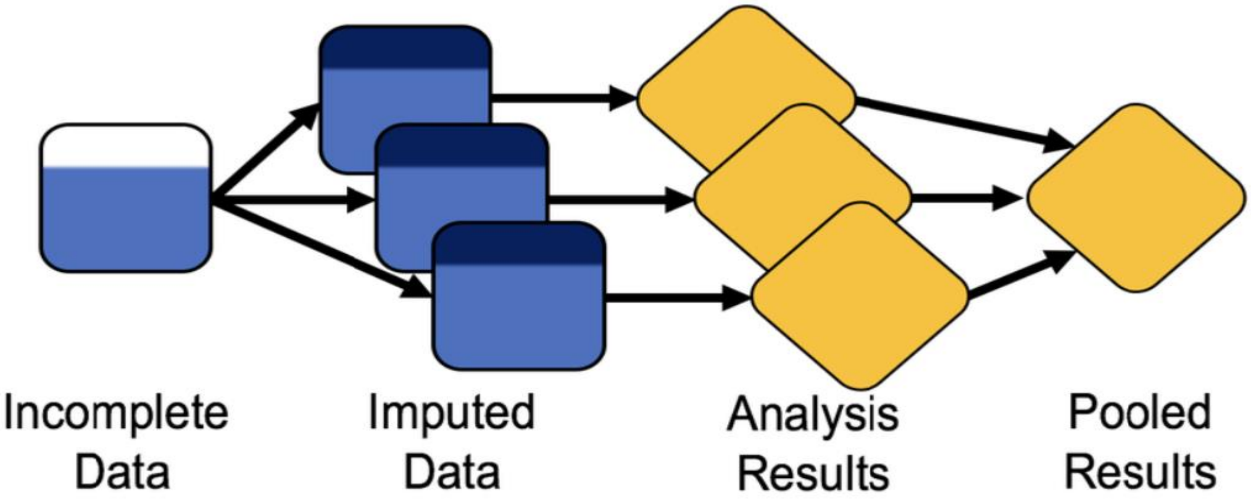
Mobile ID	Mobile Package	Download Speed	DL Speed Missing	Data Limit Usage
1	Fast+	200	0	80%
2	Lite	100	0	70%
3	Fast+	200	0	10%
4	Fast+	200	1	80%
5	Lite	50	0	70%
6	Fast+	200	0	10%
7	Fast+	200	1	95%
8	Lite	200	0	77%
9	Fast+	180	0	95%

10) Random Sampling Imputation

Random sampling imputation is in principle similar to mean/median imputation, in the sense that it aims to preserve the statistical parameters of the original variable, for which data is missing. Random sampling consists of taking a random observation from the pool of available observations of the variable, and using that randomly extracted value to fill the NA. In Random Sampling one takes as many random observations as missing values are present in the variable. Random sample imputation assumes that the data are missing completely at random (MCAR). If this is the case, it makes sense to substitute the missing values, by values extracted from the original variable distribution.

Multiple Imputation

Multiple Imputation (MI) is a statistical technique for handling missing data. The key concept of MI is to use the distribution of the observed data to estimate a set of plausible values for the missing data. Random components are incorporated into these estimated values to show their uncertainty. Multiple datasets are created and then analysed individually but identically to obtain a set of parameter estimates. Estimates are combined to obtain a set of parameter estimates. As a flexible way of handling more than one missing variable, apply a Multiple Imputation by Chained Equations (MICE) approach. The benefit of the multiple imputation is that in addition to restoring the natural variability of the missing values, it incorporates the uncertainty due to the missing data, which results in a valid statistical inference. Refer to reference section to get more information on MI and MICE. Below is a schematic representation of MICE.



Predictive/Statistical models that impute the missing data

This should be done in conjunction with some kind of cross-validation scheme in order to avoid leakage. This can be very effective and can help with the final model. There are many options for such predictive model including neural network. Here I am listing a few which are very popular.

Linear Regression

In regression imputation, the existing variables are used to make a prediction, and then the predicted value is substituted as if an actual obtained value. This approach has a number of advantages, because the imputation retains a great deal of data over the list wise or pair wise deletion and avoids significantly altering the standard deviation or the shape of the distribution. However, as in a mean substitution, while a regression imputation substitutes a value that is predicted from other variables, no novel information is added, while the sample size has been increased and the standard error is reduced.

Random Forest

Random forest is a non-parametric imputation method applicable to various variable types that works well with both data missing at random and not missing at random. Random forest uses multiple decision trees to estimate missing values and outputs OOB (out of bag) imputation error estimates. One caveat is that random forest works best with large datasets and using random forest on small datasets runs the risk of overfitting.

k-NN (k Nearest Neighbour)

k-NN imputes the missing attribute values on the basis of nearest K neighbour. Neighbours are determined on the basis of distance measure. Once K neighbours are determined, missing value are imputed by taking mean/median or mode of known attribute values of missing attribute.

Maximum likelihood

There are a number of strategies using the maximum likelihood method to handle the missing data. In these, the assumption that the observed data are a sample drawn from a multivariate normal distribution is relatively easy to understand. After the parameters are estimated using the available data, the missing data are estimated based on the parameters which have just been estimated.

Expectation-Maximization

Expectation-Maximization (EM) is a type of the maximum likelihood method that can be used to create a new data set, in which all missing values are imputed with values estimated by the maximum likelihood methods. This approach begins with the expectation step, during which the parameters (e.g., variances, co-variances, and means) are estimated, perhaps using the list wise deletion. Those estimates are then used to create a regression equation to predict the missing data. The maximization step uses those equations to fill in the missing data. The expectation step is then repeated with the new parameters, where the new regression equations are determined to “fill in” the missing data. The expectation and maximization steps are repeated until the system stabilizes.

Sensitivity analysis

Sensitivity analysis is defined as the study which defines how the uncertainty in the output of a model can be allocated to the different sources of uncertainty in its inputs. When analysing the missing data, additional assumptions on the reasons for the missing data are made, and these assumptions are often applicable to the primary analysis. However, the assumptions cannot be definitively validated for the correctness. Therefore, the National Research Council has proposed that the sensitivity analysis be conducted to evaluate the robustness of the results to the deviations from the MAR assumption.

Algorithms that Support Missing Values

Not all algorithms fail when there is missing data. There are algorithms that can be made robust to missing data, such as k-Nearest Neighbours that can ignore a column from a distance measure when a value is missing. There are also algorithms that can use the missing value as a unique and different value when building the predictive model, such as classification and regression trees.

Algorithm like XGBoost takes into consideration of any missing data. If your imputation does not work well, try a model that is robust to missing data.

Recommendations

Missing data reduces the power of a model. Some amount of missing data is expected, and the target sample size is increased to allow for it. However, such cannot eliminate the potential bias. More attention should be paid to the missing data in the design and performance of the studies and in the analysis of the resulting data. Application of the machine learning model techniques should only be performed after the maximal efforts put to reduce missing data in the design and prevention techniques.

A statistically valid analysis which has appropriate mechanisms and assumptions for the missing data strongly recommended. Most of the imputation technique can cause bias. It is difficult to know whether the multiple imputation or full maximum likelihood estimation is best, but both are superior to the traditional approaches. Both techniques are best used with large samples. In general, multiple imputation is a good approach when analysing data sets with missing data.

What is Data Cleansing and Transformation

Data creation and consumption is becoming a way of life. According to a recent IBM report, the world produced approximately 2.5 quintillion bytes of data a day in 2017. By the year 2020, analysts predict that every second, about 1.7 megabytes of new information will be created for every person on earth.

Most of this data is stored on the internet, making it the largest database on earth. Google, Amazon, Microsoft, and Facebook alone store 1,200 petabytes (1.2 million terabytes).

But using data comes with risks. The MIT Sloan Management review reported that financial losses due to bad data equal between 15%- 25% of a company’s revenue. And according to a 2018 IDC Business Analytic Solutions survey, data scientists spend 73% of their time doing the hard work of preparing data for more, value-added activities like predictive analytics or forecasting.

With so much potential for adverse business outcomes (lost time, lost sales, lost market share, lost customers, and more), businesses seeking to use data analytics to grow their bottom line really need to grasp the concepts of data cleansing and transformation.

While traditional web scraping methods may supply a large volume of data, it’s often messy and disorganized. Web data integration (WDI), however, focuses on data quality and controls. WDI has built-in Excel-like transform functions that allow you to normalize data right within the web application.

This allows you to extract, prepare, integrate, and consume data all within the same environment – from idea to insights in one simple solution. In turn, you're able to utilize data with a high level of trust and confidence and take full advantage.

Before Starting With Data Cleansing and Transformation

Oftentimes, analysts are tempted to jump into cleaning data without completing some essential tasks. The items listed below set the stage for data wrangling by helping the analyst identify all of the data elements (but only the data elements they need to address):

- **Define the business case**: Knowing the business objective is the first step toward proper data wrangling. A good business case lays out the alignment with corporate strategy, the customer problems to be solved, new or updated business processes, the estimated costs, and the projected return on investment. These parameters help identify necessary (and unnecessary) data insights.
- **Data source investigation**: A sufficiently-designed data model sheds light on the possible sources of data such as websites and webpages to populate that model. Specifically, a thorough data source examination includes:
    - Identifying the data required by the business case
    - Knowing whether the data will be integrated directly into an application or business process or if it will be used to drive an analytical investigation
    - Identifying what trends project team members anticipate seeing as web data is collected over time
    - Cataloging possible data sources and their data stewards in a mature IT environment
    - Understanding the delivery mechanism and frequency of refreshed data from the source

Also note that the value of web data increases, and over time, it becomes possible to perform time-series and trend analyses on the data. Therefore, your decision-making improves, and you gain a deeper understanding of how significant events such as a celebrity endorsement or limited-time sale impact your company.

- **Data profiling**: This step entails really getting to know the data before transforming it. Data profiling reveals data structure, null records, outliers, junk data, and potential data quality issues, etc. A thorough inspection of the data can help determine if a data source is worthy of inclusion in the data transformation effort, possible data quality issues, and the amount of wrangling required to transform the data for business analytics use.

The process of defining the business case, developing the data model, and finding and profiling data sources performs a valuable, winnowing function on data sources: it identifies only the data needed, and the processing work necessary to make that data usable. The stage is now set for data cleansing.

Data Cleansing

Only after the data source is evaluated and profiled can data cleansing proceed. Data cleansing depends on thorough and continuous data profiling to identify data quality issues that must be addressed.

Generally speaking, all applications of cleansing, transformation, profiling, discovery, wrangling, etc., should be in terms of data that is captured/extracted from the web. Each website should be treated as a source, and you should use language from that standpoint rather than the traditional ETL/data integration slant on enterprise data management and data from traditional sources.

Common data cleansing best practices can include (but are not limited to):

- **Defining a data quality plan**: Derived from the business case (see above), the quality plan may also entail some conversation with business stakeholders to tease out answers to questions like "What are our data extraction standards," "What opportunities do we have to automate the data pipeline," "What data elements are key to downstream products and processes," "Who is responsible for ensuring data quality," and "How do we determine accuracy."
- **Validating accuracy**: One type of accuracy is taking steps to ensure data is correctly entered at the point of collection – for example, if a website has changed and the value is no longer there, or if the pricing of a product is only available when you put an item into a shopping cart because of a promotion.
- **Deduplicating**: No source data set is perfect, and sometimes source systems send duplicate rows. The key here is to know the "natural key" of each record, meaning the field or fields that uniquely identify each row. If an inbound data set includes records having the same natural key, all but one of the rows could be removed.
- **Handling blank values**: Are blank values represented as "NA," "Null," "-1," or "TBD"? If so, deciding on a single value for consistency's sake will help eliminate stakeholder confusion. A more advanced approach is *imputing* values. This means using populated cells in a column to make a reasonable guess at the missing values, such as finding the average of the populated cells and assigning that to the blank cells.
- **Reformatting values**: If the source data's date fields are in the MM-DD-YYYY format, and your target date fields are in the YYYY/MM/DD format, update the source date fields to match the target format.
- **Threshold checking**: This is a more nuanced data cleansing approach. It includes comparing a current data set to historical values and record counts. For example, in the health care world, let's say a monthly claims data source averages total allowed amounts of $2M and unique claim counts of 100K. If a subsequent data load arrives with a total allowed amount of $10M and 500K unique claims, those amounts exceed the normal expected threshold of variance, and should trigger additional scrutiny.

Upfront data cleansing provides accurate and consistent data to downstream processes and analytics, which will increase customer confidence in the data. Import's WDI aids in data cleansing by preparing extracted data by exploring, assessing, and refining the data quality. It also cleanses, normalizes, and enriches the data using 100+ spreadsheet functions and formulas.

Data Wrangling

Data wrangling (sometimes called "data preparation" or "data munging") is the practice of converting cleansed data into the dimensional model for a particular business case. It involves two key components of the WDI process – extraction and preparation. The former involves rendering CSS, processing JavaScript, interpreting network traffic, etc. The latter harmonizes the data and ensures quality assurance.

Here are some data wrangling best practices:

- **Start with a small test set**: One of the challenges of Big Data is working with large data sets, especially early in the data transformation where analysts need to quickly iterate through many different exploratory techniques. To help tame the unruly beast of 500 million rows, apply random sampling to the data set to explore the data and lay out the preparation steps. This method will greatly accelerate data exploration and quickly set the stage for further transformation.
- **Understand the columns and data types**: Having a data dictionary (a document that describes a data set's column names, business definition, and data type) can really help with this step. It's necessary to ensure that the data values actually stored in a column match the business definition of that column. For example, a column called "date_of_birth" should be formatted in a format like MM/DD/YYYY. Combining this practice with data profiling, described above, should help the analyst really get to know the data.
- **Visualize source data**: Using common graphing tools and techniques can help bring the "current state" of the data to life. Histograms show distributions, scatter plots help find outliers, pie graphs show percentage to whole, and line graphs can show trends in key fields over time. Showing how data looks in visual form is also a great way to explain exploratory findings and needed transformations to non-technical users.
- **Zero in on only the needed data elements**: This is where having a well-defined business case can really help. Since most source data sets have far more columns than are actually needed, it's imperative to wrangle only the columns required by the business case. Proper application of this practice will save untold amounts of time, money, and credibility.
- **Turn it into actionable data**: The steps above shed light on the manipulations, transformations, calculations, reformatting, etc. needed to convert the web source data into the target format. A skilled analyst can create repeatable workflows that translate the required business rules into data wrangling action.
- **Test early and often**: Ideally, reliable expected values are available to test the results of a data wrangling effort. A good business case could include expected values for validation purposes. But even if not, knowing the business question and iteratively testing the results of data wrangling should help testers surface data transformation issues for resolution early in the process.

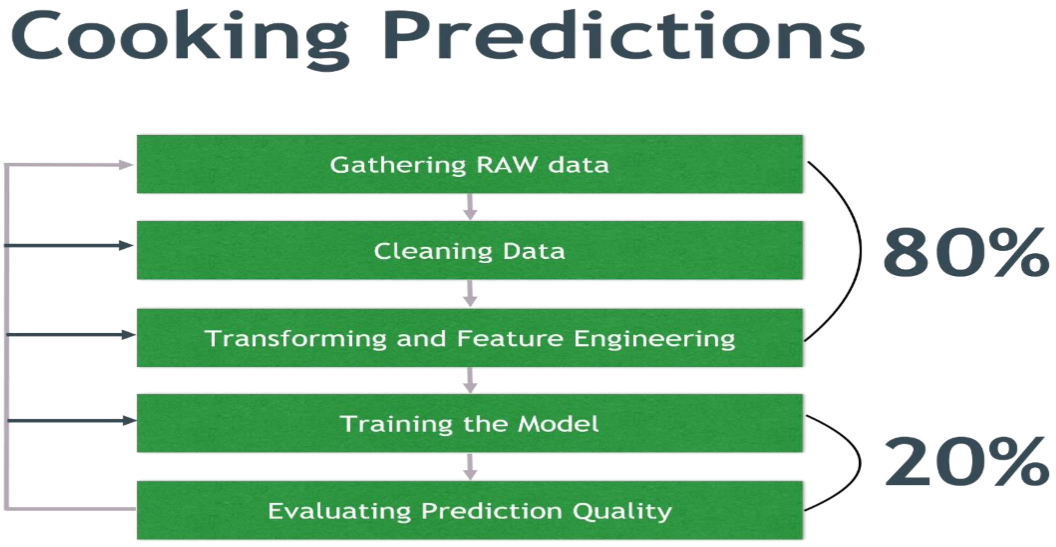
5 Importance of clean-select-transform-data

When people first see a demo of a Machine Learning product, there is a general feeling that it is something magical. Probably because it is a different way of seeing how computers work. Instead of using closed code, which behaves like a calculator that always throws the same result when the input data is identical, a Machine Learning system works with the patterns it discovers from the data as it's fed with them. These are dynamic programs that change over time and from which you may not obtain the same results over time even when the input data is the same.

Opposite of what we might call "traditional" programming, a Machine Learning-based system adjusts to a changing environment, adapting to new situations as it is fed with fresh data.

However, it is important to understand that predictive analytics is not magic, and although the algorithm learns, it can only extract valuable information from the data we provide. Algorithms do not have the same intuitive capacity as humans, whether this is good or bad, and the success of the system depends mainly on the input data.

The trick of the "demos" you can find is that the data is already selected, cleaned and transformed so the algorithm can "easily" discover the patterns. The entire process of generating a predictive model, from data capture to prediction, is equivalent in this approach to cooking a delectable dish. The ingredients would be the data, and the recipe the algorithm: if the ingredients are in bad condition, no matter how good the recipe is, the dish will not turn out well. Equivalently; if it is not good quality data (i.e. not well selected, clean and transformed), even the best algorithm will give us poor quality predictions.



From data to algorithms

Let’s take a brief look at the previous data preparation process. It is common to face a predictive analysis project with a lot of data. A lot is a lot. The first task is to collect them all. They are usually in different repositories:

- The company’s CRM.
- SQL (or non-SQL) databases.
- Spreadsheets.
- Social networks.
- In the business billing program.
- Email list management program.
- Bank transaction reports.
- In someone’s head…

Often these data are “dirty”, that means, they have errors or discrepancies between the different fields in different databases. For example, the letter “ñ” or the accents may be encoded in different formats depending on where we have collected the data. The data cleansing phase includes, among other tasks:

- Match formats.
- Discard fields.
- Correct spelling mistakes.
- Format dates.
- Remove duplicate columns.
- Delete unusable records.
- Treat missing data.

With the “clean” data you can start to select what will be useful to make your predictions. At this stage you have to keep the “signal” and remove the fields that provide “noise”. This part of the process is usually called **Feature Engineering**:

The data transformation, which also belongs to the so-called Feature Engineering, tries to generate new predictor fields based on the ones we already have knowledge of the domain (of the business, of the area being analyzed) is essential to tackle this phase. This, and the phase of selecting predictive fields, are the ones that require the most intellectual and creative effort, since it is not only necessary to know the field of study, but it is also necessary to know with a certain depth how predictive algorithms work, how they interpret the data internally and how are the relationships between them.

As an example, you might think that in a churn prediction project it is enough to have both, the acquisition and the retirement dates available. We could interpret the algorithm, by analyzing these two data, as being capable of “deducing” the client’s seniority. But that is not the case. The transformation in this case very simple: it would be to add a new field that would be the subtraction of the two dates and transform it into number of days (or months, or years, depending on whether we consider it better). A small modification like this can greatly improve the predictive capability of the system.

Conclusions

Machine Learning As A Service (MLAAS) platforms are bringing predictive analysis (as opposed to descriptive analysis) substantially closer to businesses of all sizes. What the big ones have been doing for years, is now becoming generalized to all companies. The process we are going through reminds us of the evolution of databases in the 1980s and 1990s: what was initially difficult to explain (how they work and what they were for) is now so integrated into all systems that it is difficult to find a single company that does not have a database in its core.

The algorithms are important, but they're not the most important thing. The preliminary phase of data collection and preparation requires minimal effort and knowledge in order to carry out a successful project. This phase can take between 80% and 90% of the project time.

6 Data cleansing & data transformation

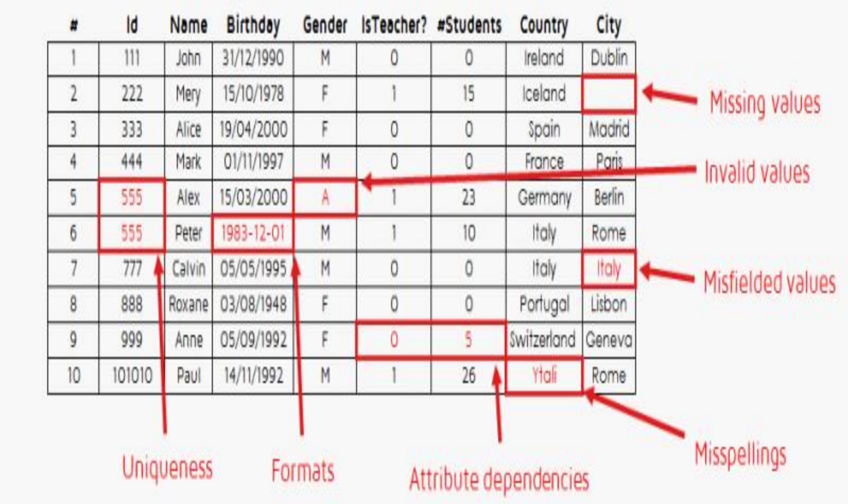
Machine Learning" or "Data Science" are trending concepts. There are different websites (like Kaggle) where Data Scientists can analyse big datasets to **resolve some real problems using Machine Learning techniques**. These techniques are applied to huge amounts of information, to learn the relationships between its features.

Machine Learning algorithms use all the values of the dataset. If we have a "dirty" dataset with a lot of mistakes and issues, these algorithms will not learn as effectively. It's therefore neccesary to fix the issues first and *then* apply the Machine Learning algorithm.

Data Cleansing

What kind of issues affect the quality of data?

- ***Invalid values***: Some datasets have well-known values, e.g. gender must only have "F" (Female) and "M" (Male). In this case it's easy to detect wrong values.
- ***Formats***: The most common issue. It's possible to get values in different formats like a name written as "Name, Surname" or "Surname, Name".
- ***Attribute dependencies***: When the value of a feature depends on the value of another feature. For example, if we have some school data, the "number of students" is related to whether the person "is teacher?". If someone is not a teacher he/she can't have any students.
- ***Uniqueness***: It's possible to find repeated data in features that only allow unique values. For example, we can't have two products with the same identifier.
- ***Missing values***: Some features in the dataset may have blank or null values.
- ***Misspellings:*** Incorrectly written values.
- ***Misfielded values***: When a feature contains the values of another.



#	Id	Name	Birthday	Gender	IsTeacher?	#Students	Country	City
1	111	John	31/12/1990	M	0	0	Ireland	Dublin
2	222	Mery	15/10/1978	F	1	15	Iceland	
3	333	Alice	19/04/2000	F	0	0	Spain	Madrid
4	444	Mark	01/11/1997	M	0	0	France	Paris
5	555	Alex	15/03/2000	A	1	23	Germany	Berlin
6	555	Peter	1983-12-01	M	1	10	Italy	Rome
7	777	Calvin	05/05/1995	M	0	0	Italy	Italy
8	888	Roxane	03/08/1948	F	0	0	Portugal	Lisbon
9	999	Anne	05/09/1992	F	0	5	Switzerland	Geneva
10	101010	Paul	14/11/1992	M	1	26	Ytali	Rome

How can I detect and fix these issues?

There are a great deal of methods that you can use to find these issues. For instance:

- ***Visualisation***: Visualising all the values of each feature, or taking a random sample to see if it's right.
- ***Outlier analysis***: Analysing if data can be a human error. E.g. a 300 year old person in the "age" feature.
- ***Validation code***: It's possible to create a code that checks if the data is right. For example, in uniqueness, checking if the length of the data is the same as the length of the vector of unique values.

We can apply many methods to fix the different issues:

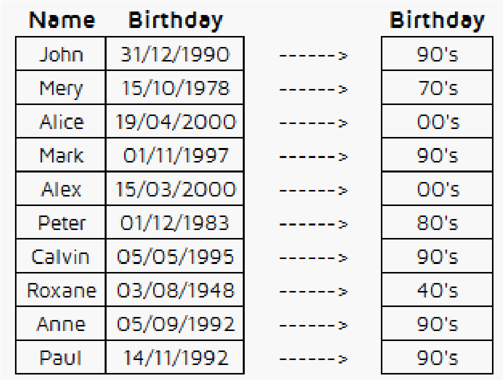
- ***Misspelled data***: Replacing incorrect fields by the most similar value in the feature.
- ***Uniqueness***: Switching one of the repeated field with another value that is not in the feature.
- ***Missing data***: Handling missing data is a key decision. We can change null values with the mean, median or mode of the feature.
- ***Formats***: Having the same number of decimals, the same format in the dates …

Data Transformation

Is it possible to transform the features to gain more information?

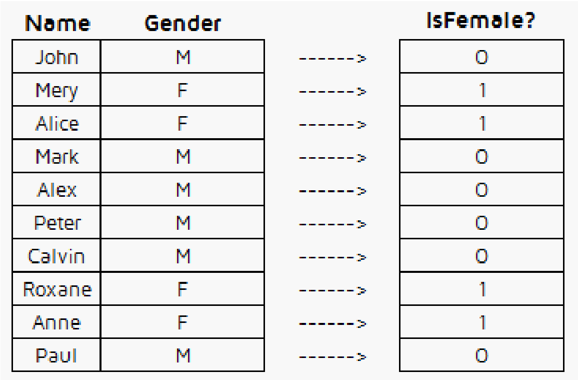
There are many methods that add information to the algorithm:

- ***Data Binning or Bucketing***: A pre-processing technique used to reduce the effects of minor observation errors. The sample is divided into intervals and replaced by categorical values.



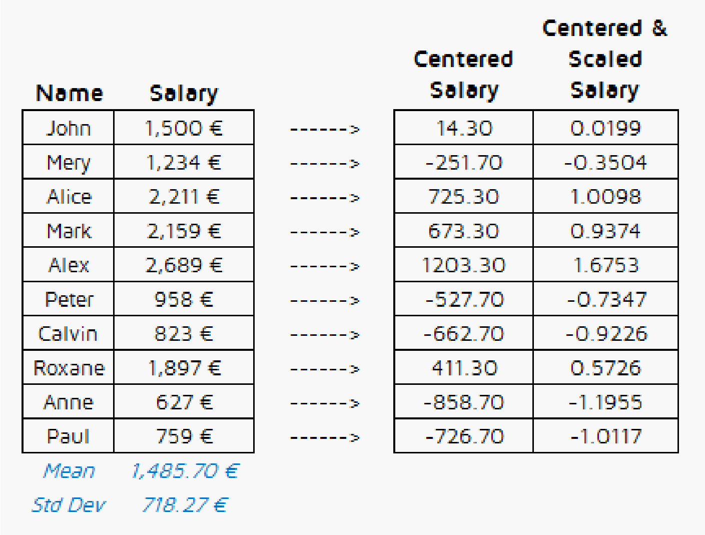
Name	Birthday		Birthday
John	31/12/1990	------>	90's
Mery	15/10/1978	------>	70's
Alice	19/04/2000	------>	00's
Mark	01/11/1997	------>	90's
Alex	15/03/2000	------>	00's
Peter	01/12/1983	------>	80's
Calvin	05/05/1995	------>	90's
Roxane	03/08/1948	------>	40's
Anne	05/09/1992	------>	90's
Paul	14/11/1992	------>	90's

- ***Indicator variables***: This technique converts categorical data into boolean values by creating indicator variables. If we have more than two values (n) we have to create n-1 columns.



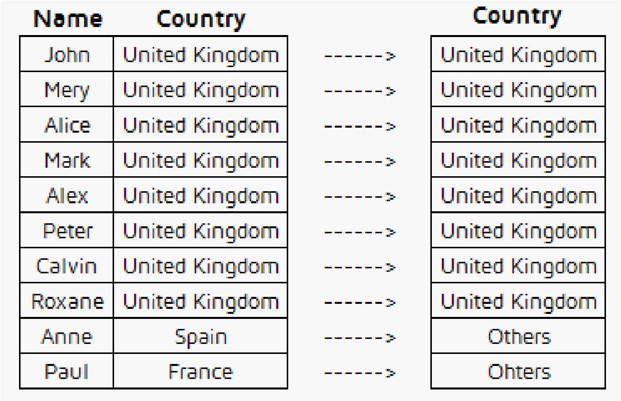
Name	Gender		IsFemale?
John	M	------>	0
Mery	F	------>	1
Alice	F	------>	1
Mark	M	------>	0
Alex	M	------>	0
Peter	M	------>	0
Calvin	M	------>	0
Roxane	F	------>	1
Anne	F	------>	1
Paul	M	------>	0

- ***Centering & Scaling***: We can centre the data of one feature by substracting the mean to all values. To scale the data, we should divide the centered feature by the standard deviation:

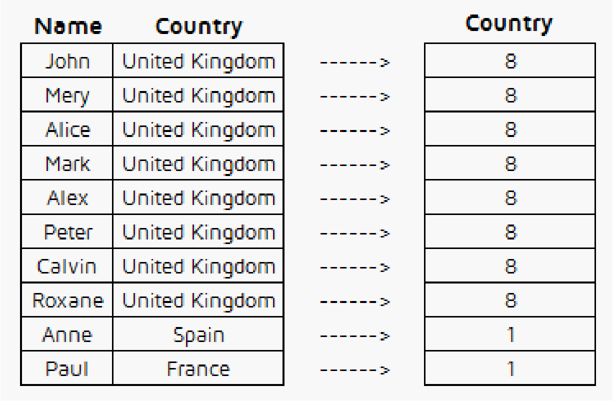


Name	Salary		Centered Salary	Centered & Scaled Salary
John	1,500 €	------>	14.30	0.0199
Mery	1,234 €	------>	-251.70	-0.3504
Alice	2,211 €	------>	725.30	1.0098
Mark	2,159 €	------>	673.30	0.9374
Alex	2,689 €	------>	1203.30	1.6753
Peter	958 €	------>	-527.70	-0.7347
Calvin	823 €	------>	-662.70	-0.9226
Roxane	1,897 €	------>	411.30	0.5726
Anne	627 €	------>	-858.70	-1.1955
Paul	759 €	------>	-726.70	-1.0117
Mean	*1,485.70 €*			
Std Dev	*718.27 €*			

- ***Other techinques***: For example, we can group the outliers with the same value or replace the value with the number of times that it appears in the feature:



Name	Country		Country
John	United Kingdom	------>	United Kingdom
Mery	United Kingdom	------>	United Kingdom
Alice	United Kingdom	------>	United Kingdom
Mark	United Kingdom	------>	United Kingdom
Alex	United Kingdom	------>	United Kingdom
Peter	United Kingdom	------>	United Kingdom
Calvin	United Kingdom	------>	United Kingdom
Roxane	United Kingdom	------>	United Kingdom
Anne	Spain	------>	Others
Paul	France	------>	Ohters



Name	Country		Country
John	United Kingdom	------>	8
Mery	United Kingdom	------>	8
Alice	United Kingdom	------>	8
Mark	United Kingdom	------>	8
Alex	United Kingdom	------>	8
Peter	United Kingdom	------>	8
Calvin	United Kingdom	------>	8
Roxane	United Kingdom	------>	8
Anne	Spain	------>	1
Paul	France	------>	1

https://www.coursera.org/lecture/data-analytics-business-capstone/data-cleanup-and-transformation-IOAXb -video

8 Data Transformation, Data Cleaning, Data Cleansing Software

- Ab Initio, provides high-performance software library and graphical environment for data transformation
- AMADEA, data Extraction, Transformation, and Real Time Reporting software
- AnalyticsCanvas, helps automate Google Analytics and Facebook insights dataflow, connects to various data sources, performs calculations and data transformations, and export data for storage and visualization.
- analytix*BASE*, a self-service analytics software for business users to quickly and easily create reports and analysis without SQL knowledge, using an intuitive and visual work-flow interface.
- Astera ReportMiner enables users with no technical background to extract & transform data from virtually any report, and map and export data anywhere.
- BioComp iManageData(tm), Accesses, cleans, filters, converts and transforms data from files, Excel, Oracle, SQL Server, process control systems and more.

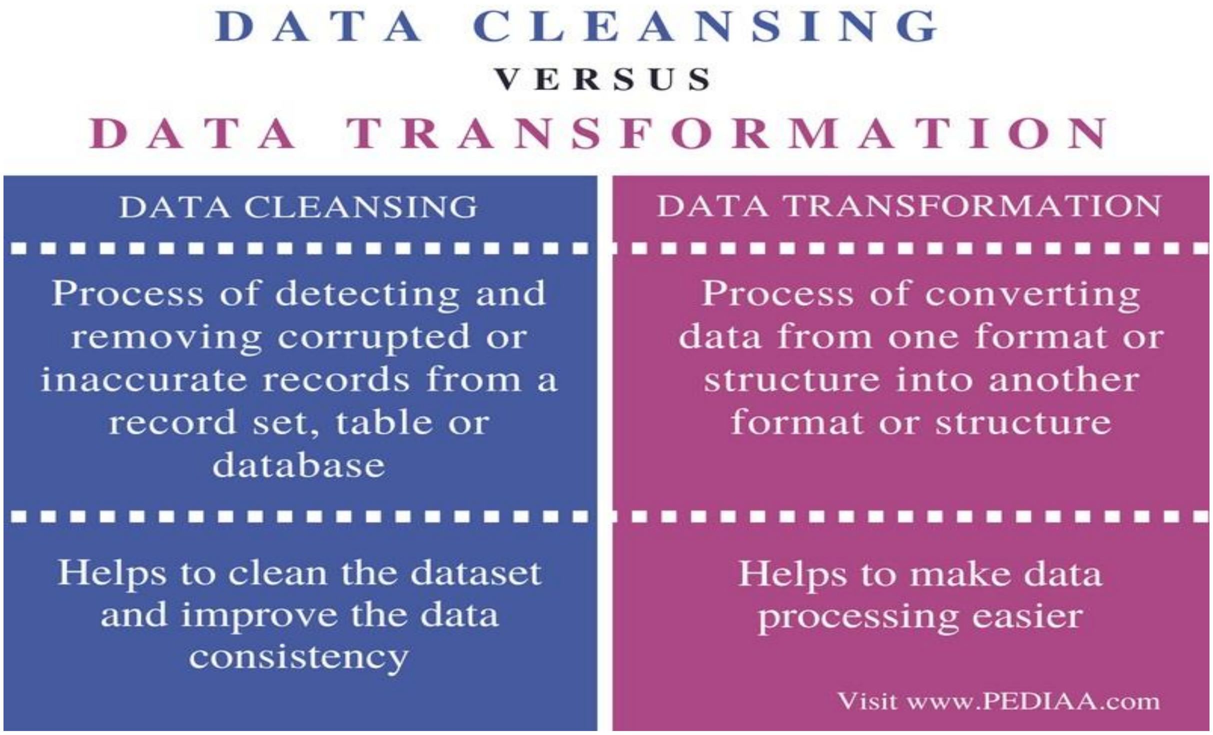
- Blendo ETL-as-a-service platform, designed to support the integration of data from multiple SaaS applications and quickly loading into a data warehouse like Google BigQuery, Amazon Redshift or Microsoft SQL Server.
- COMGEN - Disk, tape and data conversion and data recovery experts, Commercial and General Systems.
- Datadex provides an end-to-end data cleaning, data cataloging, data linking, data governance, data exchange, and data merging functionality. Our goal is Data Augmentation by leveraging existing data and increasing sample sizes or feature sets.
- Data Ladder, offering Data Matching, Profiling, deduplication, and Enrichment software and services.
- Data Manager, windows GUI application for data transformation and cleansing before data mining.
- DataFlux, provides Data Management solutions including Data profiling, Data quality, Data integration and Data augmentation
- DataPreparator, Java based tool to explore, manipulate, transform and prepare data using a graphical user interface.
- Datamartist, allows large amounts of data from multiple sources to be combined together, enhanced and repaired without the need for database development.
- Datatect, a powerful program for generating realistic test data to ASCII flat files or directly to RDBMS including Oracle, Sybase, SQL Server, and Informix.
- Dataskope, department-level tools to map, transform, alarm, output and view high volumes of binary or ASCII input data.
- DQ Now, profiling, cleansing, and dedup tools, providing a clear view of the data
- DQ Global, data cleansing, data management software, including de-duplication, merge/purge, address correction and suppression.
- Easy Data Transform, with easy data blending, cleaning and reformatting for Windows and Mac.
- FreeSight avoid "spreadsheet hell" with patented tools to simplify and automate data blending, cleansing, analysis and reporting.
- GritBot, for identifying anomalies in data (compatible with See5 and Cubist).
- Hummingbird ETL, powerful data integration solution.
- MiningMart platform, for the preparation of relational data for Knowledge Discovery, free for research and non-commercial applications.
- MoData technology platform aggregates, cleanses and generates analytic cubes from disparate ERP and CRP sources and provides a data science and insights delivery platform.
- NewView from SPSS
- OpenRefine (ex-Google Refine), a powerful tool for working with messy data, cleaning it, transforming it from one format into another, extending it with web services, and linking it to databases like Freebase.
- Optimus, a Python framework for cleansing, preparing and exploratory data analysis in a distributed fashion with Apache Spark (Pyspark).
- proMISS, imputes missing values in databases.
- Relational Tools streamline application testing by allowing moving, editing and comparing referentially intact sets of complex relational data.
- Sagent, provides a suite of data transformation and loading tools

- The Software Bureau, providing Cygnus and SwiftSort innovative data quality software.
- Syncsort, fast high-volume sorting, filtering, reformatting, aggregating, and more
- The TrueData COMponent, functions to programmatically standardise your data, process it phonetically, and output a match key.
- WinPure, powerful data cleaning software, including duplication removal, email suggestions, statistics and more.

9 Difference Between Data Cleansing and Data Transformation

the **main difference** between data cleansing and data transformation is that **the data cleansing is the process of removing the unwanted data from a dataset or database while the data transformation is the process of converting data from one format to another format.**

A business organization stores data in different data sources. It is important to make decisions by analyzing the data. Analyzing data from multiple data sources is difficult. Therefore, business organizations use data warehouses. It is a central location that stores consolidated data from multiple databases. Data warehouses help to create reports, analyze data, visualize data and make valuable business decisions. In other words, data warehousing supports the overall business intelligence process. Data cleansing and data transformation are two techniques that are used in data warehousing. Data cleansing refers to eliminating meaningless data from the data set to improve data consistency while data transformation refers to converting data from one structure to another structure to make them easier for processing.



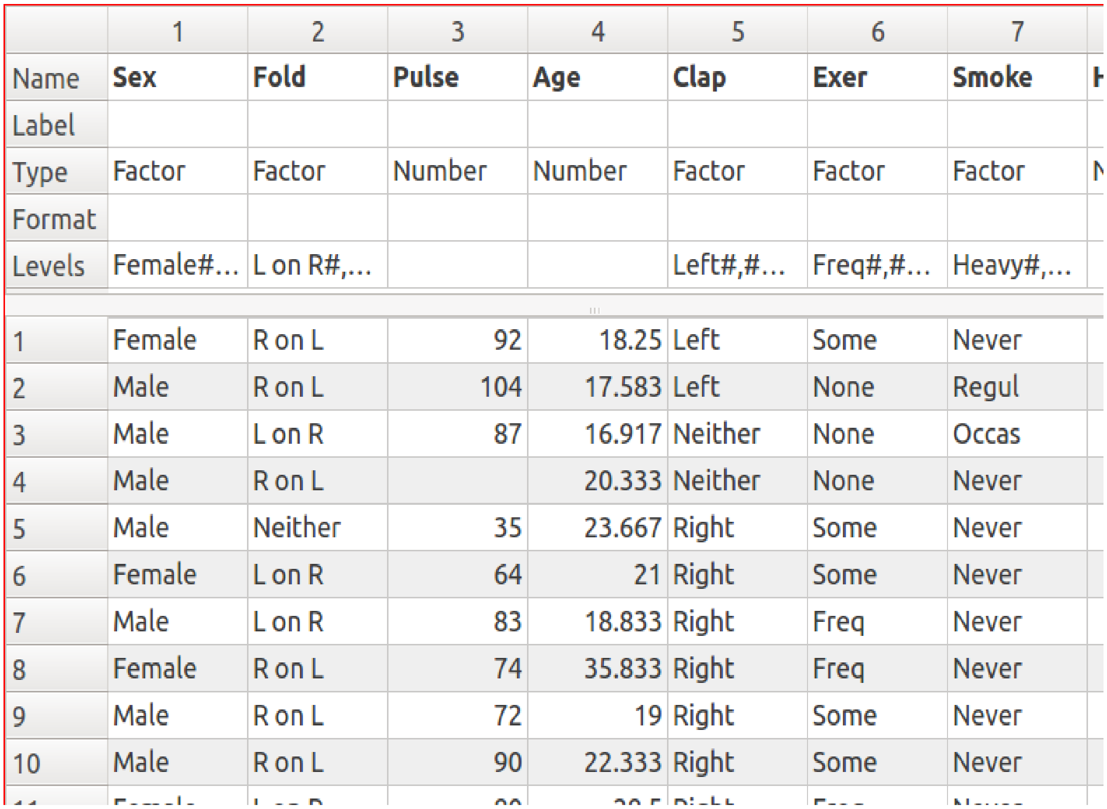
DATA CLEANSING
VERSUS
DATA TRANSFORMATION

DATA CLEANSING	DATA TRANSFORMATION
Process of detecting and removing corrupted or inaccurate records from a record set, table or database	Process of converting data from one format or structure into another format or structure
Helps to clean the dataset and improve the data consistency	Helps to make data processing easier

Visit www.PEDIAA.com

What is Data Cleansing

A business organization uses various sources to store data. They can have different databases such as Oracle, MySQL, etc. It is difficult to analyze data in different data sources. Data warehousing provides a solution to this issue. It helps to collect, store and manage data from a variety of data sources into a central location called a data warehouse. The data warehouse gets data from transactional systems and various relational databases. Finally, this data is processed and analyzed to get meaningful business insights.



	1	2	3	4	5	6	7	
Name	**Sex**	**Fold**	**Pulse**	**Age**	**Clap**	**Exer**	**Smoke**	**H**
Label								
Type	Factor	Factor	Number	Number	Factor	Factor	Factor	N
Format								
Levels	Female#...	L on R#,...			Left#,#...	Freq#,#...	Heavy#,...	
1	Female	R on L	92	18.25	Left	Some	Never	
2	Male	R on L	104	17.583	Left	None	Regul	
3	Male	L on R	87	16.917	Neither	None	Occas	
4	Male	R on L		20.333	Neither	None	Never	
5	Male	Neither	35	23.667	Right	Some	Never	
6	Female	L on R	64	21	Right	Some	Never	
7	Male	L on R	83	18.833	Right	Freq	Never	
8	Female	R on L	74	35.833	Right	Freq	Never	
9	Male	R on L	72	19	Right	Some	Never	
10	Male	R on L	90	22.333	Right	Some	Never	

Figure 1: Dataset

The data should be cleaned and transformed before loading into the warehouse. The extracted data from multiple sources can consist of meaningless data. Dummy values, contradictory data, absence of data are considered as meaningless data. These unnecessary data must be removed from the dataset. Overall, data cleaning will not just provide a clean dataset. It also brings data consistency to different sets of data that have merged from various data sources.

What is Data Transformation

After cleansing, the data is transformed into a suitable format. Data transformation helps to process the data easily. Data transforming can be simple or complex depending on the required changes on the data. Standardizing data, character set conversion, encoding handling, splitting or merging fields, conversion units of measurements into a standard format, aggregation, consolidation, delete duplicate data are some of the tasks involved in data transformation.

After completing the data transformation, the data is loaded into the data warehouse for processing. Finally, the senior management and data analysts can take decisions based on the processed data. Apart from data warehousing, data cleansing and data transforming are also used for statistical and mathematical operations.

Difference Between Data Cleansing and Data Transformation

Definition

Data cleansing is the process of detecting and removing corrupted or inaccurate records from a record set, table or database while the data transformation is the process of converting data from one format or structure into another format or structure.

Usage

Furthermore, data cleansing helps to clean the dataset and improve the data consistency while data transformation helps to make data processing easier.