These days almost anything can be a valuable source of information. The primary challenge lies in extracting the insights from the said information and making sense of it, which is the point of Big Data. However, you also need to prep the data first, which is Data Wrangling in a nutshell.

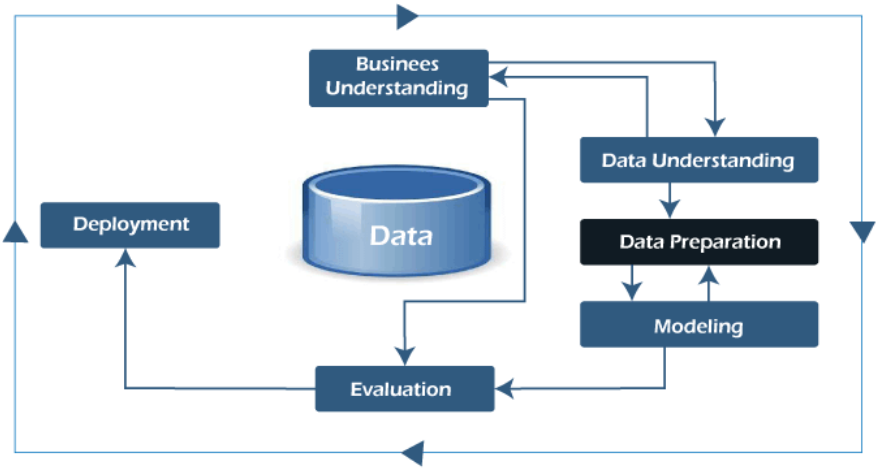
The nature of the information is that it requires a certain kind of organization to be adequately assessed. This process requires a crystal clear understanding of which operations need what sort of data. Let's look closer at wrangling data and explain why it is so important.

Data Wrangling:

Sometimes, data Wrangling is referred to as ***data munging***. It is the process of transforming and mapping data from one "raw" data form into another format to make it more appropriate and valuable for various downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data.

The process of data wrangling may include further munging, data visualization, data aggregation, training a statistical model, and many other potential uses. Data wrangling typically follows a set of general steps, which begin with extracting the raw data from the data source, "munging" the raw data (e.g., sorting) or parsing the data into predefined data structures, and finally depositing the resulting content into a data sink for storage and future use.

Wrangling the data is usually accompanied by Mapping. The term "Data Mapping" refers to the element of the wrangling process that involves identifying source data fields to their respective target data fields. While Wrangling is dedicated to transforming data, Mapping is about connecting the dots between different elements.



Importance of Data Wrangling

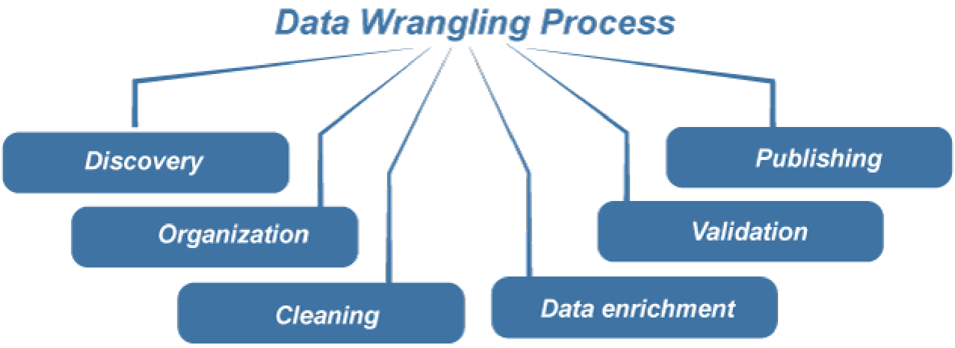
Some may question if the amount of work and time devoted to data wrangling is worth the effort. A simple analogy will help you understand. The foundation of a skyscraper is expensive and time-consuming before the above-ground structure starts. Still, this solid foundation is extremely valuable for the building to stand tall and serve its purpose for decades. Similarly, once the code and infrastructure foundation are gathered for data handling, it will deliver immediate results (sometimes almost instantly) for as long as the process is relevant. However, skipping necessary data wrangling steps will lead to significant downfalls, missed opportunities, and erroneous models that damage the reputation of analysis within the organization.

Data wrangling software has become an indispensable part of data processing. The primary importance of using data wrangling tools can be described as follows:

- Making raw data usable. Accurately wrangled data guarantees that quality data is entered into the downstream analysis.
- Getting all data from various sources into a centralized location so it can be used.
- Piecing together raw data according to the required format and understanding the business context of data.
- Automated data integration tools are used as data wrangling techniques that clean and convert source data into a standard format that can be used repeatedly according to end requirements. Businesses use this standardized data to perform crucial, cross-data set analytics.
- Cleansing the data from the noise or flawed, missing elements.
- Data wrangling acts as a preparation stage for the data mining process, which involves gathering data and making sense of it.
- Helping business users make concrete, timely decisions.

Data Wrangling Process

Data Wrangling is one of those technical terms that are more or less self-descriptive. The term "wrangling" refers to rounding up information in a certain way. This operation includes a sequence of the following processes:



1. **Discovery:** Before starting the wrangling process, it is critical to think about what may lie beneath your data. It is crucial to think critically about what results from you anticipate from your data and what you will use it for once the wrangling process is complete. Once you've determined your objectives, you can gather your data.
2. **Organization:** After you've gathered your raw data within a particular dataset, you must structure your data. Due to the variety and complexity of data types and sources, raw data is often overwhelming at first glance.
3. **Cleaning:** When your data is organized, you can begin cleaning your data. Data cleaning involves removing outliers, formatting nulls, and eliminating duplicate data. It is important to note that cleaning data collected from web scraping methods might be more tedious than cleaning data collected from a database. Essentially, web data can be highly unstructured and require more time than structured datafrom a database.
4. **Data enrichment:** This step requires that you take a step back from your data to determine if you have enough data to proceed. Finishing the wrangling process

without enough data may compromise insights gathered from further analysis. For example, investors looking to analyze product review data will want a significant amount of data to portray the market and increase investment intelligence

5. **Validation:** After determining you gathered enough data, you will need to apply validation rules to your data. Validation rules, performed in repetitive sequences, confirm that your data is consistent throughout your dataset. Validation rules will also ensure quality as well as security. This step follows similar logic utilized in data normalization, a data standardization process involving validation rules.
6. **Publishing:** The final step of the data munging process is data publishing. Data publishing involves preparing the data for future use. This may include providing notes and documentation of your wrangling process and creating access for other users and applications.

Data Wrangling Tools

There are different tools for data wrangling that can be used for gathering, importing, structuring, and cleaning data before it can be fed into analytics and BI apps. You can use automated tools for data wrangling, where the software allows you to validate data mappings and scrutinize data samples at every step of the transformation process. This helps to detect and correct errors in data mapping quickly.

Automated data cleaning becomes necessary in businesses dealing with exceptionally large data sets. The data team or data scientist is responsible for Wrangling manual data cleaning processes. However, in smaller setups, non-data professionals are responsible for cleaning data before leveraging it.

Various data wrangling methods range from munging data with scripts to spreadsheets. Additionally, with some of the more recent all-in-one tools, everyone utilizing the data can access and utilize their data wrangling tools. Here are some of the more common data wrangling tools available.

- Spreadsheets / Excel Power Query is the most basic manual data wrangling tool.
- OpenRefine - An automated data cleaning tool that requires programming skills
- Tabula

  It is a tool suited for all data types

  1. Google DataPrep

     It is a data service that explores, cleans, and prepares data

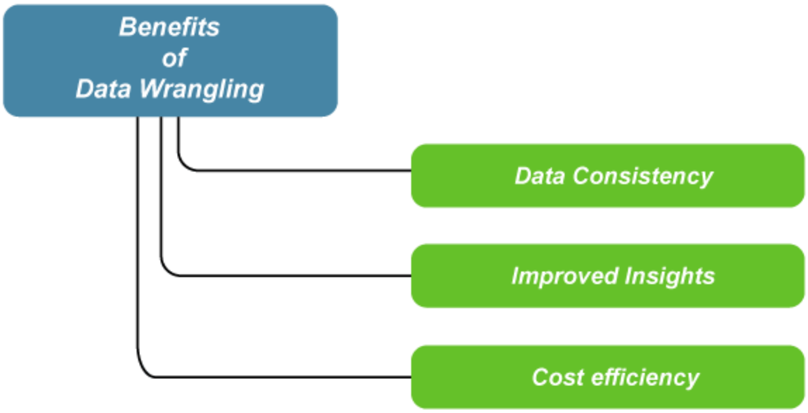
     1. Data wrangler

        It is a data cleaning and transforming tool

        1. Plotly (data wrangling with Python) is useful for maps and chart data.
        2. CSVKit converts data.

Benefits of Data Wrangling

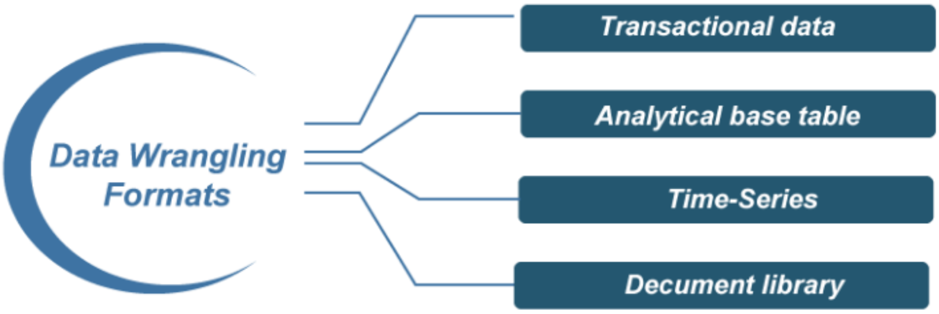
As previously mentioned, big data has become an integral part of business and finance today. However, the full potential of said data is not always clear. Data processes, such as data discovery, are useful for recognizing your data's potential. But to fully unleash the power of your data, you will need to implement data. Here are some of the key benefits of data wrangling.



- **Data consistency:** The organizational aspect of data wrangling offers a resulting dataset that is more consistent. Data consistency is crucial for business operations that involve collecting data input by consumers or other human end-users. For example, if a human-end user submits personal information incorrectly, such as making a duplicate customer account, which would consequently impact further performance analysis.
- **Improved insights:** Data wrangling can provide statistical insights about metadata by transforming the metadata to be more constant. These insights are often the result of increased data consistency, as consistent metadata allows automated tools to analyze the data faster and more accurately. Particularly, if one were to build a model regarding projected market performance, data wrangling would clean the metadata to allow your model to run without any errors.
- **Cost efficiency:** As previously mentioned, because data-wrangling allows for more efficient data analysis and model-building processes, businesses will ultimately save money in the long run. For instance, thoroughly cleaning and organizing data before sending it off for integration will reduce errors and save developers time.
    - Data wrangling helps to improve data usability as it converts data into a compatible format for the end system.
    - It helps to quickly build data flows within an intuitive user interface and easily schedule and automate the data-flow process.
    - Integrates various types of information and sources (like databases, web services, files, etc.)
    - Help users to process very large volumes of data easily and easily share data-flow techniques.

Data Wrangling Formats

Depending on the type of data you are using, your final result will fall into four final formats: de-normalized transactions, analytical base table (ABT), time series, or document library. Let's take a closer look at these final formats, as understanding these results will inform the first few steps of the data wrangling process, which we discussed above.



- **Transactional data:** Transactional data refers to business operation transactions. This data type involves detailed subjective information about particular transactions, including client documentation, client interactions, receipts, and notes regarding any external transactions.
- **Analytical Base Table (ABT):** Analytical Base Table data involves data within a table with unique entries for each attribute column. ABT data is the most common business data type as it involves various data types that contribute to the most common data sources. Even more notable is that ABT data is primarily used for AI and ML, which we will examine later.
- **Time-series:** Time series data involves data that has been divided by a particular amount of time or data that has a relation with time, particularly sequential time. For example, tracking data regarding an application's downloads over a year or tracking traffic data over a month would be considered time series data.
- **Document library:** Lastly, document library data is information that involves a large amount of textual data, particularly text within a document. While document libraries contain rather massive amounts of data, automated data mining tools specifically designed for text mining can help extract entire texts from documents for further analysis.

Data Wrangling Examples

Data wrangling techniques are used for various use cases. The most commonly used examples of data wrangling are for:

1. Merging several data sources into one data set for analysis
2. Identifying gaps or empty cells in data and either filling or removing them
3. Deleting irrelevant or unnecessary data
4. Identifying severe outliers in data and either explaining the inconsistencies or deleting them to facilitate analysis

Businesses also use data wrangling tools to

- Detect corporate fraud
- Support data security
- Ensure accurate and recurring data modeling results
- Ensure business compliance with industry standards
- Perform Customer Behavior Analysis
- Reduce time spent on preparing data for analysis
- Promptly recognize the business value of your data
- Find out data trends

Data Acquisition:

Data acquisition is the process of sampling signals that measure real-world physical conditions and converting the resulting samples into digital numeric values that a computer can manipulate.

Data acquisition systems (DAS or DAQ) convert physical conditions of analog waveforms into digital values for further storage, analysis, and processing.

In simple words, Data Acquisition is composed of two words: Data and Acquisition, where data is the raw facts and figures, which could be structured and unstructured and acquisition means acquiring data for the given task at hand.

Data acquisition meaning is to collect data from relevant sources before it can be stored, cleaned, preprocessed, and used for further mechanisms. It is the process of retrieving relevant business information, transforming the data into the required business form, and loading it into the designated system.

A data scientist spends 80 percent of the time searching, cleaning, and processing data. With Machine Learning becoming more widely used, some applications do not have enough labeled data. Even the best Machine Learning algorithms cannot function properly without good data and cleaning of the data. Also, Deep learning techniques require vast amounts of data, as, unlike Machine Learning, these techniques automatically generate features. Otherwise, we would have garbage in and garbage out. Hence, data acquisition or collection is a very critical aspect.

The data acquisition in Data Science involves:

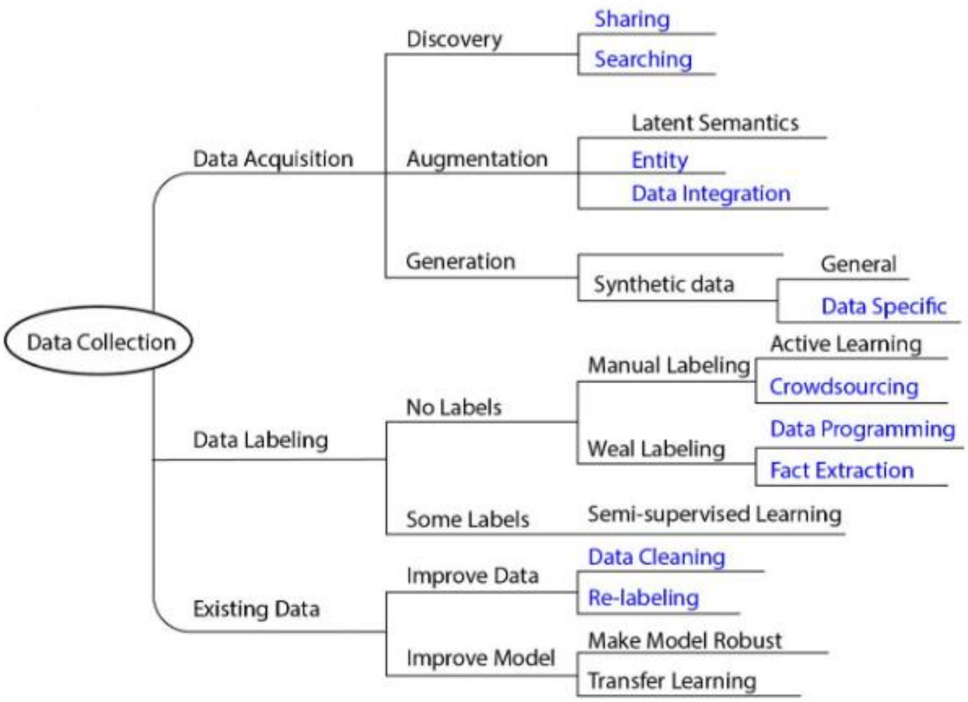
- **Collection and Integration of the data:** The data is extracted from various sources and also the data is usually available at different places so the multiple data needs to be combined to be used. The data acquired is typically in raw format and not suitable for immediate consumption and analysis. This calls for future processes such as:
- **Formatting:** Prepare or organize the datasets as per the analysis requirements.
- **Labeling:** After gathering data, it is required to label the data. One such instance is in an application factory, one would want to label the images of the components if the components are defective or not. In another case, if constructing a knowledge base by extracting information from the web then would need to label that it is implicitly assumed to be true. At times, it is needed to manually label the data.

The Data Acquisition Process

The process of data acquisition involves searching for the datasets that can be used to train the Machine Learning models. Having said that, it is not simple. There are various approaches to acquiring data, here have bucketed into three main segments such as:

1. Data Discovery
2. Data Augmentation
3. Data Generation

Each of these has further sub-processes depending upon their functionality. The figure below lays out an overview of the research landscape of data collection for machine learning. We’ll dive deep into each of these.



1. **Data Discovery:**
   The first approach to acquiring data is Data discovery. It is a key step when indexing, sharing, and searching for new datasets available on the web and incorporating data lakes. It can be broken into two steps: Searching and Sharing. Firstly, the data must be labeled or indexed and published for sharing using many available collaborative systems for this purpose.
2. **Data Augmentation:**
   The next approach for data acquisition is Data augmentation. Augment means to make something greater by adding to it, so here in the context of data acquisition, we are essentially enriching the existing data by adding more external data. In Deep and Machine learning, using pre-trained models and embeddings is common to increase the features to train on.
3. **Data Generation:**
   As the name suggests, the data is generated. If we do not have enough and any external data is not available, the option is to generate the datasets manually or automatically. Crowd sourcing is the standard technique for manual construction of the data where people are assigned tasks to collect the required data to form the generated dataset. There are automatic techniques available as well to generate synthetic datasets. Also, the data generation method can be seen as data augmentation when there is data available however it has missing values that need to be imputed.

Data Acquisition Tools

- **Data Warehouses and ETL**

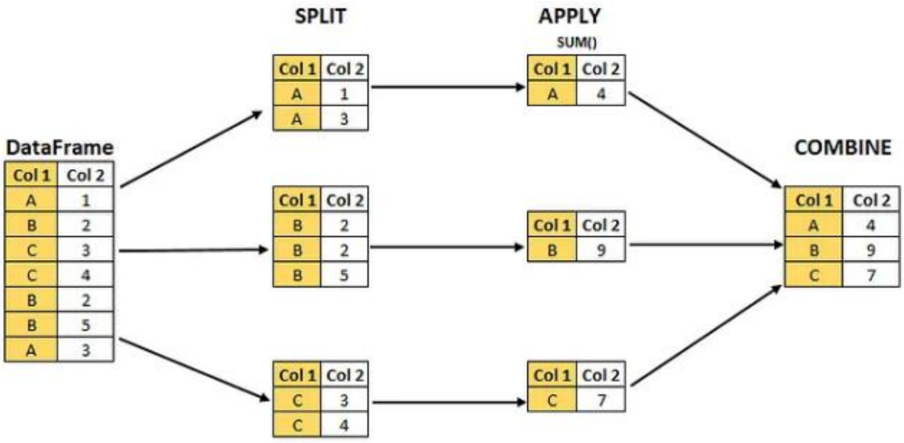
  ETL, or extract, transform, and load, is a procedure used in data warehousing. During this procedure, data is extracted from multiple data source systems, transformed in the staging area, and loaded into the Data Warehouse system.
- **Data Lakes and ELT**

A Data Lake should be constructed if ELT is to be used. Data is extracted, typically using physical files, loaded into your data lake on your cloud storage, and only then is the data transformed and cleaned.

- **Cloud Data Warehouse Providers**
  A cloud data warehouse is a managed service database prepared for scalable business intelligence and analytics in a public cloud.

split-apply-combine paradigm:

split-apply-combine strategy in which we break up a big problem into small manageable pieces (Split), operate on each piece independently (Apply) and then put all the pieces back together (Combine). Split-Apply-Combine can be used by many existing tools by using GroupBy function in SQL and Python, LOD in Tableau, and by using plyr functions in R to name a few. In this article, we will not be discussing only the implementation of this strategy, but also we will see some relevant application of this strategy in Feature Engineering.



Col 1	Col 2
A	1
B	2
C	3
C	4
B	2
B	5
A	3

Col 1	Col 2
A	1
A	3

Col 1	Col 2
B	2
B	2
B	5

Col 1	Col 2
C	3
C	4

Col 1	Col 2
A	4

Col 1	Col 2
B	9

Col 1	Col 2
C	7

Col 1	Col 2
A	4
B	9
C	7

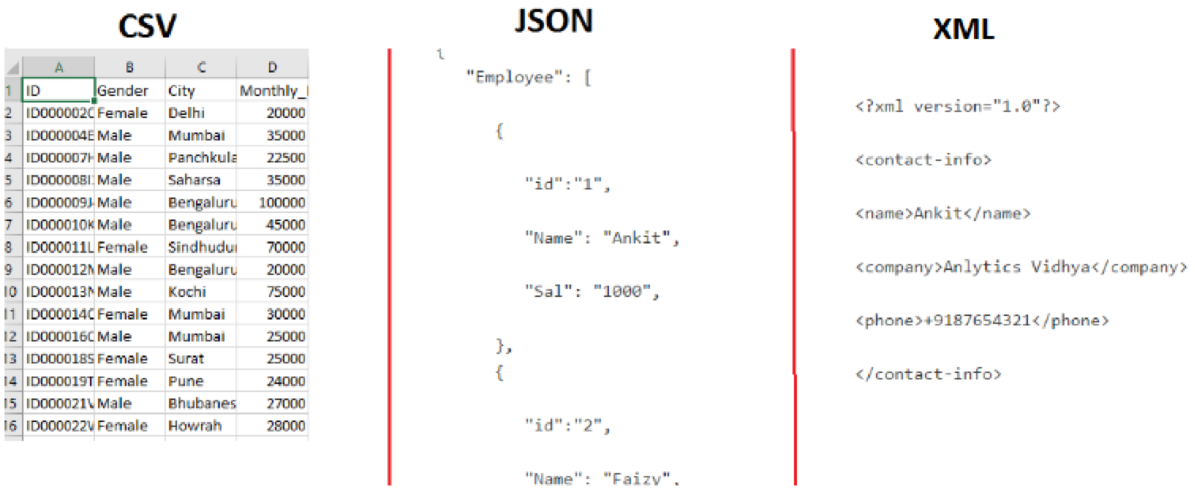
1. **Split:** Split the data into groups based on some criteria thereby creating a GroupBy object. (We can use the column or a combination of columns to split the data into groups)
2. **Apply:** Apply a function to each group independently. (Aggregate, Transform, or Filter the data in this step)
3. **Combine:** Combine the results into a data structure (Pandas Series, Pandas DataFrame)

Data Formats:

Data appears in different sizes and shapes, it can be numerical data, text, multimedia, research data, or a few other types of data. The data format is said to be a kind of format which is used for coding the data. The data is coded in different ways. It is being coded, so that it can be read, recognized, and used by the different applications and programs.

A file format is a standard way in which information is encoded for storage in a file. First, the file format specifies whether the file is a binary or ASCII file. Second, it shows how the information is organized. For example, comma-separated values (CSV) file format stores tabular data in plain text.

To identify a file format, you can usually look at the file extension to get an idea. For example, a file saved with name “Data” in “CSV” format will appear as “Data.csv”. By noticing “.csv” extension we can clearly identify that it is a “CSV” file and data is stored in a tabular format.



CSV

	A	B	C	D
1	ID	Gender	City	Monthly_I
2	ID000002C	Female	Delhi	20000
3	ID000004E	Male	Mumbai	35000
4	ID000007H	Male	Panchkula	22500
5	ID000008I	Male	Saharsa	35000
6	ID000009J	Male	Bengaluru	100000
7	ID000010K	Male	Bengaluru	45000
8	ID000011L	Female	Sindhudu	70000
9	ID000012N	Male	Bengaluru	20000
10	ID000013N	Male	Kochi	75000
11	ID000014C	Female	Mumbai	30000
12	ID000016C	Male	Mumbai	25000
13	ID000018S	Female	Surat	25000
14	ID000019T	Female	Pune	24000
15	ID000021V	Male	Bhubanes	27000
16	ID000022V	Female	Howrah	28000

JSON

```
{
  "Employee": [

     {

        "id":"1",

        "Name": "Ankit",

        "Sal": "1000",

     },
     {

        "id":"2",

        "Name": "Faizy",
```

XML

```
<?xml version="1.0"?>

<contact-info>

<name>Ankit</name>

<company>Anlytics Vidhya</company>

<phone>+9187654321</phone>

</contact-info>
```

Reading the data from CSV in Python

Let us look at how to read a CSV file in Python. For loading the data you can use the “pandas” library in python.

```
import pandas as pd
df = pd.read_csv("train.csv")
```

Above code will load the train.csv file in DataFrame df.

Reading the data from XLSX file

Let’s load the data from XLSX file and define the sheet name. For loading the data you can use the Pandas library in python.

```
import pandas as pd
df = pd.read_excel("train.xlsx", sheetname = "Invoice")
```

Above code will load the sheet “Invoice” from “train.xlsx” file in DataFrame df.

Reading the data from JSON file

JavaScript Object Notation(JSON) is a text-based open standard designed for exchanging the data over web. JSON format is used for transmitting structured data over the web. The JSON file format can be easily read in any programming language because it is language-independent data format.

Let’s take an example of a JSON file

The following example shows how a typical JSON file stores information of employees.

```
{
  "Employee": [
    {
      "id":"1",          "Name": "Ankit",      "Sal": "1000",
    },
    {
      "id":"2",          "Name": "Faizy",      "Sal": "2000",
    }
  ]
```

}

Reading a JSON file

Let’s load the data from JSON file. For loading the data you can use the pandas library in python.

```
import pandas as pd
df = pd.read_json("train.json")
```

XML file format

XML is also known as Extensible Markup Language. As the name suggests, it is a markup language. It has certain rules for encoding data. XML file format is a human-readable and machine-readable file format. XML is a self-descriptive language designed for sending information over the internet. XML is very similar to HTML, but has some differences. For example, XML does not use predefined tags as HTML.

Let’s take the simple example of XML File format.

The following example shows an xml document that contains the information of an employee.

```
<?xml version="1.0"?>
      <contact-info>
            <name>Ankit</name>
            <company>Anlytics Vidhya</company>
            <phone>+9187654321</phone>
      </contact-info>
```

The “<?xml version=”1.0″?>” is a XML declaration at the start of the file (it is optional). In this deceleration, version specifies the XML version and encoding specifies the character encoding used in the document. <contact-info> is a tag in this document. Each XML-tag needs to be closed.

Reading XML in python

For reading the data from XML file you can import xml.etree. ElementTree library.

Let’s import an xml file called train and print its root tag.

```
import xml.etree.ElementTree as ET
tree = ET.parse('train.xml')
root = tree.getroot()
print(root.tag)
```

Data Imputation:

Data imputation is a method for retaining the majority of the dataset's data and information by substituting missing data with a different value. These methods are employed because it would be impractical to remove data from a dataset each time. Additionally, doing so would substantially reduce the dataset's size, raising questions about bias and impairing analysis.

Now that we learned what Data imputation is, let us see why exactly it is important.

We employ imputation since missing data can lead to the following problems:

- Distorts Dataset: Large amounts of missing data can lead to anomalies in the variable distribution, which can change the relative importance of different categories in the dataset.
- Unable to work with the majority of machine learning-related Python libraries: When utilizing ML libraries (SkLearn is the most popular), mistakes may occur because there is no automatic handling of these missing data.
- Impacts on the Final Model: Missing data may lead to bias in the dataset, which could affect the final model's analysis.
- Desire to restore the entire dataset: This typically occurs when we don't want to lose any (or any more) of the data in our dataset because all of it is crucial. Additionally, while the dataset is not very large, eliminating a portion of it could have a substantial effect on the final model.

Data Imputation Techniques

After learning about what data imputation is and its importance, we will now learn about some of the various data imputation techniques.

These are some of the data imputation techniques that we will be discussing in-depth:

1. Next or Previous Value
2. K Nearest Neighbors
3. Maximum or Minimum Value
4. Missing Value Prediction
5. Most Frequent Value
6. Average or Linear Interpolation
7. (Rounded) Mean or Moving Average or Median Value
8. Fixed Value

1. Next or Previous Value

For time-series data or ordered data, there are specific imputation techniques. These techniques take into consideration the dataset's sorted structure, wherein nearby values are likely more comparable than far-off ones. The next or previous value inside the time series is typically substituted for the missing value as part of a common method for imputed incomplete data in the time series. This strategy is effective for both nominal and numerical values.

2. K Nearest Neighbors

The objective is to find the k nearest examples in the data where the value in the relevant feature is not absent and then substitute the value of the feature that occurs most frequently in the group.

3. Maximum or Minimum Value

You can use the minimum or maximum of the range as the replacement cost for missing values if you are aware that the data must fit within a specific range [minimum, maximum] and if you are aware from the process of data collection that the measurement instrument stops recording and the message saturates further than one of such boundaries. For instance, if a price cap has been reached in a financial exchange and the exchange procedure has indeed been halted, the missing price can be substituted with the exchange boundary's minimum value.

4. Missing Value Prediction

Using a machine learning model to determine the final imputation value for characteristic x based on other features is another popular method for single imputation. The model is trained using the values in the remaining columns, and the rows in feature x without missing values are utilized as the training set. Depending on the type of feature, we can employ any regression or classification model in this situation.

In resistance training, the algorithm is used to forecast the most likely value of each missing value in all samples. A basic imputation approach, such as the mean value, is used to temporarily impute all missing values when there is missing data in more than a feature field. Then, one column's values are restored to missing. After training, the model is used to complete the missing variables. In this manner, an is trained for every feature that has a missing value up until a model can impute all of the missing values.

5. Most Frequent Value

The most frequent value in the column is used to replace the missing values in another popular technique that is effective for both nominal and numerical features.

6. Average or Linear Interpolation

The average or linear interpolation, which calculates between the previous and next accessible value and substitutes the missing value, is similar to the previous/next value imputation but only applicable to numerical data. Of course, as with other operations on ordered data, it is crucial to accurately sort the data in advance, for example, in the case of time series data, according to a timestamp.

7. (Rounded) Mean or Moving Average or Median Value

Median, Mean, or rounded mean are further popular imputation techniques for numerical features. The technique, in this instance, replaces the null values with mean, rounded mean, or median values determined for that feature across the whole dataset. It is advised to utilize the median rather than the mean when your dataset has a significant number of outliers.

8. Fixed Value

Fixed value imputation is a universal technique that replaces the null data with a fixed value and is applicable to all data types. You can impute the null values in a survey using "not answered" as an example of using fixed imputation on nominal features.

Cleaning and Munging:

Data Cleaning:

Data cleaning is fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

Steps of Data Cleaning

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 cleaning your data, such as:

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, minimize distraction from your primary target, and create a more manageable and performable dataset.

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" in any sheet, but they should be analyzed in the same category.

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, the appearance of an outlier will prove a theory you are working on. And 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.

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, such as:

- You can drop observations with missing values, but this will drop or lose information, so be careful before removing it.
- You can input missing values based on other observations; again, there is an opportunity to lose the integrity of the data because you may be operating from assumptions and not actual observations.
- You might alter how the data is used to navigate null values effectively.

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, such as:

- 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 for your next theory?
- If not, is that because of a data quality issue?

Because of incorrect or noisy data, false conclusions 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 study. 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 strategy.

Methods of Data Cleaning

There are many data cleaning methods through which the data should be run. The methods are described below:

1. **Ignore the tuples:** This method is not very feasible, as it only comes to use when the tuple has several attributes is has missing values.
2. **Fill the missing value:** This approach is also not very effective or feasible. Moreover, it can be a time-consuming method. In the approach, one has to fill in the missing value. This is usually done manually, but it can also be done by attribute mean or using the most probable value.
3. **Binning method:** This approach is very simple to understand. The smoothing of sorted data is done using the values around it. The data is then divided into several segments of equal size. After that, the different methods are executed to complete the task.
4. **Regression:** The data is made smooth with the help of using the regression function. The regression can be linear or multiple. Linear regression has only one independent variable, and multiple regressions have more than one independent variable.
5. **Clustering:** This method mainly operates on the group. Clustering groups the data in a cluster. Then, the outliers are detected with the help of clustering. Next, the similar values are then arranged into a "group" or a "cluster".

Process of Data Cleaning

The following steps show the process of data cleaning in data mining.

1. **Monitoring the errors:** Keep a note of suitability where the most mistakes arise. It will make it easier to determine and stabilize false or corrupt information. Information is especially necessary while integrating another possible alternative with established management software.
2. **Standardize the mining process:** Standardize the point of insertion to assist and reduce the chances of duplicity.
3. **Validate data accuracy:** Analyze and invest in data tools to clean the record in real-time. Tools used Artificial Intelligence to better examine for correctness.
4. **Scrub for duplicate data:** Determine duplicates to save time when analyzing data. Frequently attempted the same data can be avoided by analyzing and investing in separate data erasing tools that can analyze rough data in quantity and automate the operation.
5. **Research on data:** Before this activity, our data must be standardized, validated, and scrubbed for duplicates. There are many third-party sources, and these Approved & authorized parties sources can capture information directly from our databases. They help us to clean and compile the data to ensure completeness, accuracy, and reliability for business decision-making.

6. **Communicate with the team:** Keeping the group in the loop will assist in developing and strengthening the client and sending more targeted data to prospective customers.

Data Munging:

Data munging is the general procedure for transforming data from erroneous or unusable forms, into useful and use-case-specific ones. Without some degree of munging, whether performed by automated systems or specialized users, data cannot be ready for any kind of downstream consumption.

The term ‘Mung’ was coined in the late 60s as a somewhat derogatory term for actions and transformations which progressively degrade a dataset, and quickly became tied to the backronym “Mash Until No Good”

The data munging process: An overview

With the wide variety of verticals, use-cases, types of users, and systems utilizing enterprise data today, the specifics of munging can take on myriad forms.

1. **Data exploration:** Munging usually begins with data exploration. Whether an analyst is merely peeking at completely new data in initial data analysis (IDA), or a data scientist begins the search for novel associations in existing records in exploratory data analysis (EDA), munging always begins with some degree of data discovery.
2. **Data transformation:** Once a sense of the raw data’s contents and structure have been established, it must be transformed to new formats appropriate for downstream processing. This step involves the pure data scientist, for example un-nesting hierarchical JSON data, denormalizing disparate tables so relevant information can be accessed from one place, or reshaping and aggregating time series data to the dimensions and spans of interest.
3. **Data enrichment:** Optionally, once data is ready for consumption, data mungers might choose to perform additional enrichment steps. This involves finding external sources of information to expand the scope or content of existing records. For example, using an open-source weather data set to add daily temperature to an ice-cream shop’s sales figures.
4. **Data validation:** The final, perhaps most important, munging step is validation. At this point, the data is ready to be used, but certain common-sense or sanity checks are critical if one wishes to trust the processed data. This step allows users to discover typos, incorrect mappings, problems with transformation steps, even the rare corruption caused by computational failure or error.

Rescaling:

Rescaling data is multiplying each member of a data set by a constant term k; that is to say, transforming each number x to f(X), where f(x) = kx, and k and x are both real numbers. Rescaling will change the spread of your data as well as the position of your data points.

Your preprocessed data may contain attributes with a mixtures of scales for various quantities such as dollars, kilograms and sales volume.

Many machine learning methods expect or are more effective if the data attributes have the same scale. Two popular data scaling methods are **normalization** and **standardization.**

Data Normalization:

- Normalization refers to rescaling real valued numeric attributes into the range 0 and 1.
- It is useful to scale the input attributes for a model that relies on the magnitude of values, such as distance measures used in k-nearest neighbors and in the preparation of coefficients in regression.
- The example below demonstrate data normalization of the Iris flowers dataset.

```
from sklearn.datasets import load_iris
from sklearn import preprocessing
# load the iris dataset
iris = load_iris()
print(iris.data.shape)
# separate the data from the target attributes
X = iris.data
y = iris.target
# normalize the data attributes
normalized_X = preprocessing.normalize(X)
```

Data Standardization

- Standardization refers to shifting the distribution of each attribute to have a mean of zero and a standard deviation of one (unit variance).
- It is useful to standardize attributes for a model that relies on the distribution of attributes such as Gaussian processes.
- The example below demonstrate data standardization of the Iris flowers dataset.

```
from sklearn.datasets import load_iris
from sklearn import preprocessing
# load the Iris dataset
iris = load_iris()
print(iris.data.shape)
# separate the data and target attributes
X = iris.data
y = iris.target
# standardize the data attributes
standardized_X = preprocessing.scale(X)
```

Dimensionality Reduction:

Dimensionality reduction is a technique used to reduce the number of features in a dataset while retaining as much of the important information as possible. In other words, it is a process of transforming high-dimensional data into a lower-dimensional space that still preserves the essence of the original data.

In machine learning, high-dimensional data refers to data with a large number of features or variables. The curse of dimensionality is a common problem in machine learning, where the performance of the model deteriorates as the number of features increases. This is because the complexity of the model increases with the number of features, and it becomes more difficult to find a good solution. In addition, high-dimensional data can also lead to overfitting, where the model fits the training data too closely and does not generalize well to new data.

Dimensionality reduction can help to mitigate these problems by reducing the complexity of the model and improving its generalization performance. There are two main approaches to dimensionality reduction: feature selection and feature extraction.

Feature Selection:

Feature selection involves selecting a subset of the original features that are most relevant to the problem at hand. The goal is to reduce the dimensionality of the dataset while retaining the most important features. There are several methods for feature selection, including filter methods, wrapper methods, and embedded methods. Filter methods rank the features based on their relevance to the target variable, wrapper methods use the model performance as the criteria for selecting features, and embedded methods combine feature selection with the model training process.

Feature Extraction:

Feature extraction involves creating new features by combining or transforming the original features. The goal is to create a set of features that captures the essence of the original data in a lower-dimensional space. There are several methods for feature extraction, including principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (t-SNE). PCA is a popular technique that projects the original features onto a lower-dimensional space while preserving as much of the variance as possible.

Methods of Dimensionality Reduction

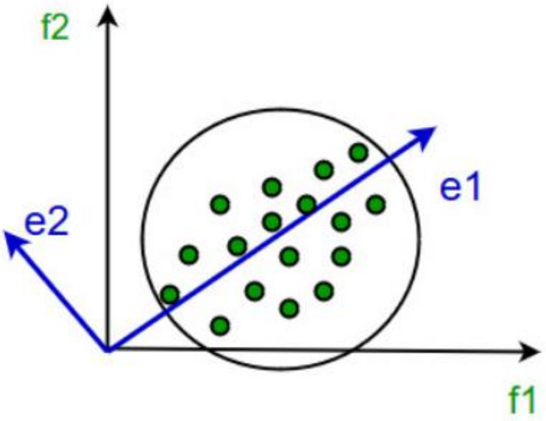
The various methods used for dimensionality reduction include:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)

Dimensionality reduction may be both linear and non-linear, depending upon the method used. The prime linear method, called Principal Component Analysis, or PCA, is discussed below.

Principal Component Analysis

This method was introduced by Karl Pearson. It works on the condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.



It involves the following steps:

- Construct the covariance matrix of the data.
- Compute the eigenvectors of this matrix.
- Eigenvectors corresponding to the largest eigenvalues are used to reconstruct a large fraction of variance of the original data.

Hence, we are left with a lesser number of eigenvectors, and there might have been some data loss in the process. But, the most important variances should be retained by the remaining eigenvectors.

PCA in python

Step 1: We will import the libraries.

```
import numpy as nmp
import matplotlib.pyplot as mpltl
import pandas as pnd
```

Step 2: We will import the dataset (wine.csv)

First, we will import the dataset and distribute it into X and Y components for data analysis.

```
DS = pnd.read_csv('Wine.csv')
X = DS.iloc[: , 0:13].values
Y = DS.iloc[: , 13].values
```

Step 3: In this step, we will split the dataset into the training set and testing set.

```
from sklearn.model_selection import train_test_split as tts
X_train, X_test, Y_train, Y_test = tts(X, Y, test_size = 0.2, random_state
= 0)
```

Step 4: Now, we will Feature Scaling.

In this step, we will do the re-processing on the training and testing set, for example, fitting the standard scale.

```
from sklearn.preprocessing import StandardScaler as SS
SC = SS()
X_train = SC.fit_transform(X_train)
X_test = SC.transform(X_test)
```

Step 5: Then, Apply the PCA function

We will apply the PCA function into the training set and testing set for analysis.

```
from sklearn.decomposition import PCA
PCa = PCA (n_components = 1)
X_train = PCa.fit_transform(X_train)
X_test = PCa.transform(X_test)
explained_variance = PCa.explained_variance_ratio_
```

Step 6: Now, we will fit Logistic Regression for the training set

```
from sklearn.linear_model import LogisticRegression as LR
classifier_1 = LR (random_state = 0)
classifier_1.fit(X_train, Y_train)
```

Advantages of Dimensionality Reduction

- It helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.

- Improved Visualization: High dimensional data is difficult to visualize, and dimensionality reduction techniques can help in visualizing the data in 2D or 3D, which can help in better understanding and analysis.
- Overfitting Prevention: High dimensional data may lead to overfitting in machine learning models, which can lead to poor generalization performance. Dimensionality reduction can help in reducing the complexity of the data, and hence prevent overfitting.
- Feature Extraction: Dimensionality reduction can help in extracting important features from high dimensional data, which can be useful in feature selection for machine learning models.
- Data Preprocessing: Dimensionality reduction can be used as a preprocessing step before applying machine learning algorithms to reduce the dimensionality of the data and hence improve the performance of the model.
- Improved Performance: Dimensionality reduction can help in improving the performance of machine learning models by reducing the complexity of the data, and hence reducing the noise and irrelevant information in the data.

Disadvantages of Dimensionality Reduction

- It may lead to some amount of data loss.
- PCA tends to find linear correlations between variables, which is sometimes undesirable.
- PCA fails in cases where mean and covariance are not enough to define datasets.
- We may not know how many principal components to keep- in practice, some thumb rules are applied.
- Interpretability: The reduced dimensions may not be easily interpretable, and it may be difficult to understand the relationship between the original features and the reduced dimensions.
- Overfitting: In some cases, dimensionality reduction may lead to overfitting, especially when the number of components is chosen based on the training data.
- Sensitivity to outliers: Some dimensionality reduction techniques are sensitive to outliers, which can result in a biased representation of the data.
- Computational complexity: Some dimensionality reduction techniques, such as manifold learning, can be computationally intensive, especially when dealing with large datasets.