

Data Munging

Unit 2

What is Data Munging?

- a.k.a **Data Wrangling**, process of processes designed to transform raw data into more readily used formats.
- The exact methods differ from project to project depending on the data you're leveraging and the goal you're trying to achieve.
- Any analyses a business performs will ultimately be constrained by the data that informs them.
- If data is incomplete, unreliable, or faulty, then analyses will be too—diminishing the value of any insights gleaned.
- Data Munging seeks to remove that risk by ensuring data is in a reliable state before it's analyzed and leveraged.

What is Data Munging? (contd.)

- It's important to note that data wrangling can be time-consuming and taxing on resources, particularly when done manually.
- This is why many organizations institute policies and best practices that help employees streamline the data cleanup process—for example, requiring that data include certain information or be in a specific format before it's uploaded to a database.

Understanding Data Munging through Analogy



Harvest / Buy Vegetables
(Import)



Wash vegetables
(Cleaning)



Chop vegetables
(Transformation)

Data Quality

- **Data Quality** refers to the state of data that determine how well suited the data is to meet the purpose.
- It is a measure of the condition of data in terms of
 - Data accuracy
 - Completeness
 - Consistency
 - Timeliness
 - Uniqueness
 - Integrity

Data Quality (contd.)

- **Data accuracy:** Not a false Information - How accurate the data is?

E.g. In resume, developer wrote 10 years of experience on Mobile App development using Flutter.

- **Completeness:** *Comprehensiveness or wholeness of data* – Do we have complete information?

All data fields contains data required for analysis.

Data Quality (contd.)

Ways to ensure data completeness

1. Determine which information is critical
2. Make certain fields required
3. Use data profiling techniques
4. Assemble a data quality team
5. Use automation and AI technologies
6. Use the right data source

Data Quality (contd.)

- **Consistency:** *Data change across all the related source; different version of data exists across different sources – Is data valid and accurate?*
- **Timeliness:** *Data belongs to the same time period as required*

E.g. You are assigned to predict future sales from today. But data for last 2 years is not so consistent due to newer system replacement.

Data Quality (contd.)

- **Uniqueness:** *No duplicate record*

ID	Organizer	Date	Test Location	Meeting	Meeting With
1	John Smith	20-01-2021	Birmingham	Evening	Rosina
2	John Doe	20-01-2021	Mexico	Evening	Rosina
3	John Doe	20-01-2021	Mexico	Evening	Rosina
4	John Doe	20-01-2021	Mexico	Evening	Rosina
5	John Doe	20-01-2021	Mexico	Evening	Rosina
6	John Smith	20-02-2021	Birmingham	Evening	Rosina

- **Conformity:** *Data follows standard format – Is data in format we require?*

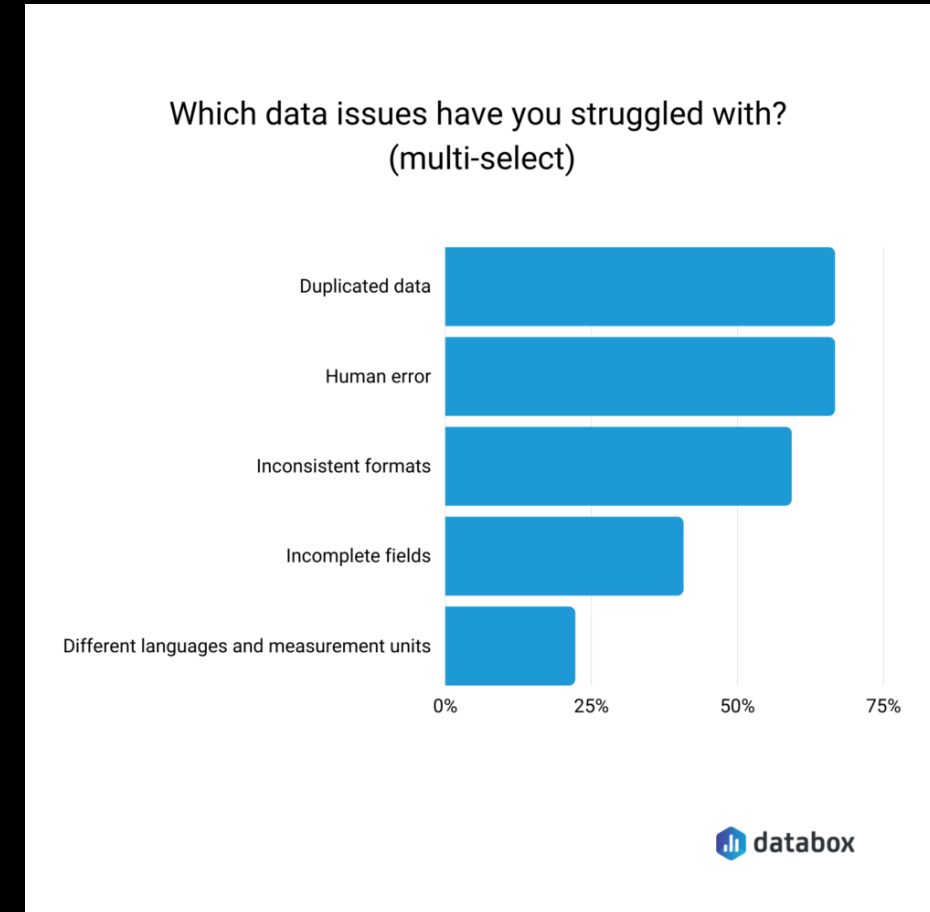
E.g. Sales data should follow 2023-03-24T17:03:41+01:00 format i.e we also require time data with date.

Common Issues with Real World Data

1. Duplicates
2. Missing Data
3. Non-Standard Data
4. Unit Mismatch

Others,

5. Data Ambiguity
6. Hidden data
7. Human Error
8. Inconsistent Data
9. Data Overload



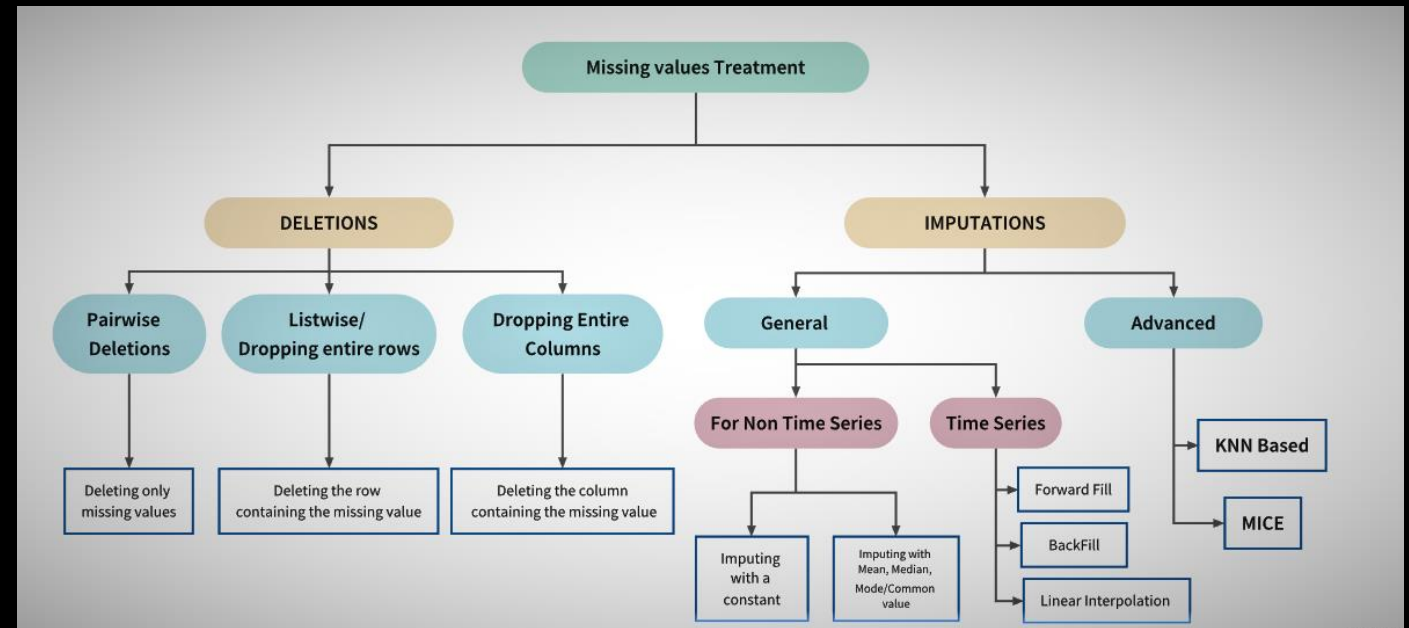
Ways to Clean Data

1. Dealing with Missing Values
2. Data Enrichment
3. Data Standardization/ Normalization

Ways to Clean Data (contd.)

1. Dealing with Missing Data

- Remove missing record
- Imputation
 - Fill with mean, mode or median
 - constant value
- Apply Algorithms or ML Techniques



<https://ml-concepts.com/2021/10/07/handling-missing-values-in-a-dataset/>

Ways to Clean Data (contd.)

2. Data Enrichment

- Data Enrichment is the process used to enhance, refine or improve the data quality.
- Data enrichment is also a process that takes raw data points and merges them with similar data points in a larger database.
- We enhance the existing information by supplementing missing or incomplete data with relevant context obtained from additional sources.
- This is achieved by merging third-party data from an external authoritative source with an existing database of first-party customer data.

Ways to Clean Data (contd.)

2. Data Enrichment (contd.)

- Businesses carry out Data Enrichment to improve the information they currently have so they can make better-informed decisions. Apart from that, it helps businesses perform the following operations:
 - Define and manage hierarchies.
 - Create new business rules for labeling and sorting data on the fly.
 - Investigate and process data that is multilingual and multi-structured.
 - Process text and Semi-structured Data more efficiently.
 - Reduce costs and optimize sales.
 - Perform Predictive Analysis.

Ways to Clean Data (contd.)

2. Data Enrichment (contd.)

- Why do we need to Enrich Data?

(In the context of business)

#1 — Enriched Data Enhances Your Personalization Strategy

#2 — Enriched Data Produces Greater Customer Insights

#3 — Enriched Data Empowers You to Hone in on VIPs

#4 — Enriched Data Enhances Campaign Performance

#5 — Enriched Data Leads to Better Decision Making

Ways to Clean Data (contd.)

2. Data Enrichment (contd.)

Techniques for Data Enrichment

- Web Scrapping
- Manual Research
- Data Append
- Data Segmentation
- Data Imputation
- Entity Extraction
- Data Categorization
- Extrapolation
- Data Correction
- Data Augmentation

Ways to Clean Data (contd.)

3. Data Standardization / ~~Normalization~~

Source 1					
Name	Email Address	Phone Number	DOB	Gender	Residential Address
John Oneel	john.neal@gmail.com	5164659494	14/2/1987	M	11400 W Olimpic BL # 200

Source 2						
First Name	Last Name	Email Address	Phone Number	DOB	Gender	Residential Address
Johnathan	O'neal	john.neal_gmail.com	+1 516-465-9494	2/14/1987	Male	11400 W Olympic 200

Ways to Clean Data (contd.)

3. Data Standardization / ~~Normalization~~ (contd.)

In the example above, you can see the following types of inconsistencies:

- **Structural**: The first source covers Customer Name as a single field, while the second one stores it as two fields – First and Last Name.
- **Pattern**: The first source has a valid email pattern enforced on the email address field, while the second one is visibly missing the @ symbol.
- **Data type**: The first source only allows digits in the Phone Number field, while the second one has a string type field that contains symbols and spaces as well.
- **Format**: The first source has date of birth in the format MM/DD/YYYY, while the second one has it in the format DD/MM/YYYY.
- **Domain value**: The first source allows Gender value to be stored as M or F, while the second source stores the complete form – Male or Female.

Ways to Clean Data (contd.)

3. Data Standardization ~~/Normalization~~ (contd.)

- Why do we need data standard definition?

A data standard definition helps identify:

- The data assets crucial to your business process,
- The necessary data fields of those assets,
- The data type, format, and pattern their values must conform to,
- The range of acceptable values for these fields, and so on.

Ways to Clean Data (contd.)

3. ~~Data Standardization~~ / Normalization

- Data normalization consists of remodeling numeric columns to a standard scale.
- Data normalization is generally considered the development of clean data. :
- Goal of the Data Normalization
 - Data normalization is the organization of data to appear similar across all records and fields.
 - It increases the cohesion of entry types, leading to cleansing, lead generation, segmentation, and higher quality data.

Ways to Clean Data (contd.)

- ~~Data Standardization~~ / Normalization

Techniques

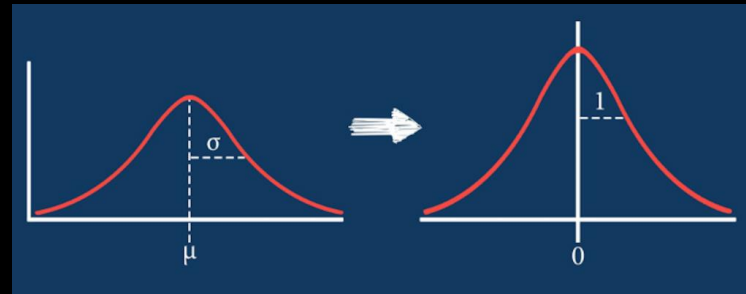
a) Min-Max Normalization

$$v' = \frac{v - \min_F}{\max_F - \min_F} (\text{new_max}_F - \text{new_min}_F) + \text{new_min}_F ,$$

a) Z-score Normalization

$$Z = \frac{x - \mu}{\sigma}$$

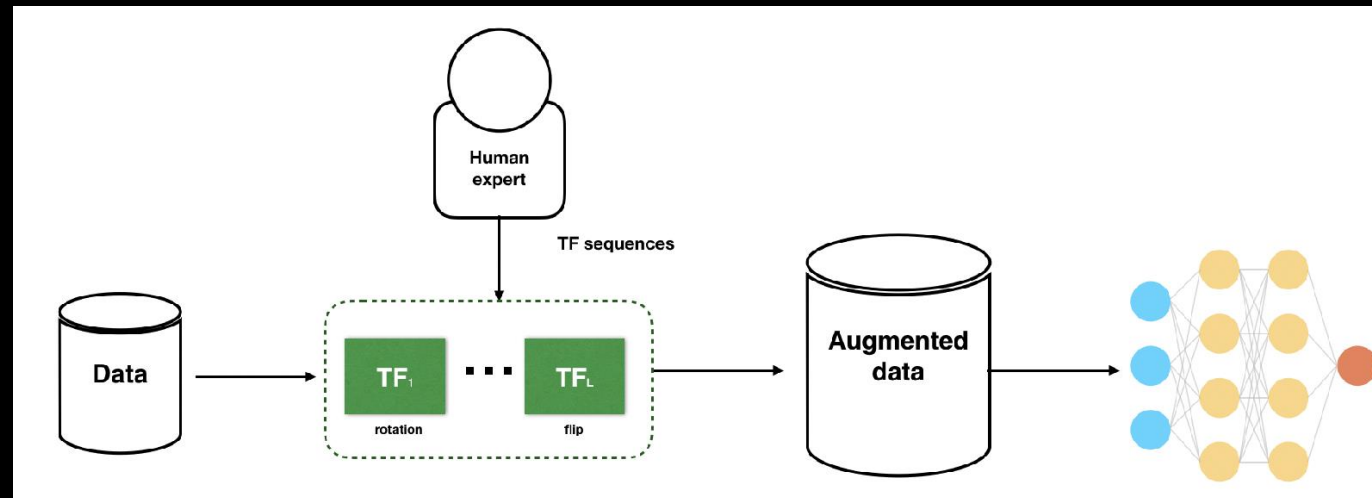
Score x
 Mean μ
 SD σ



Ways to Clean Data (contd.)

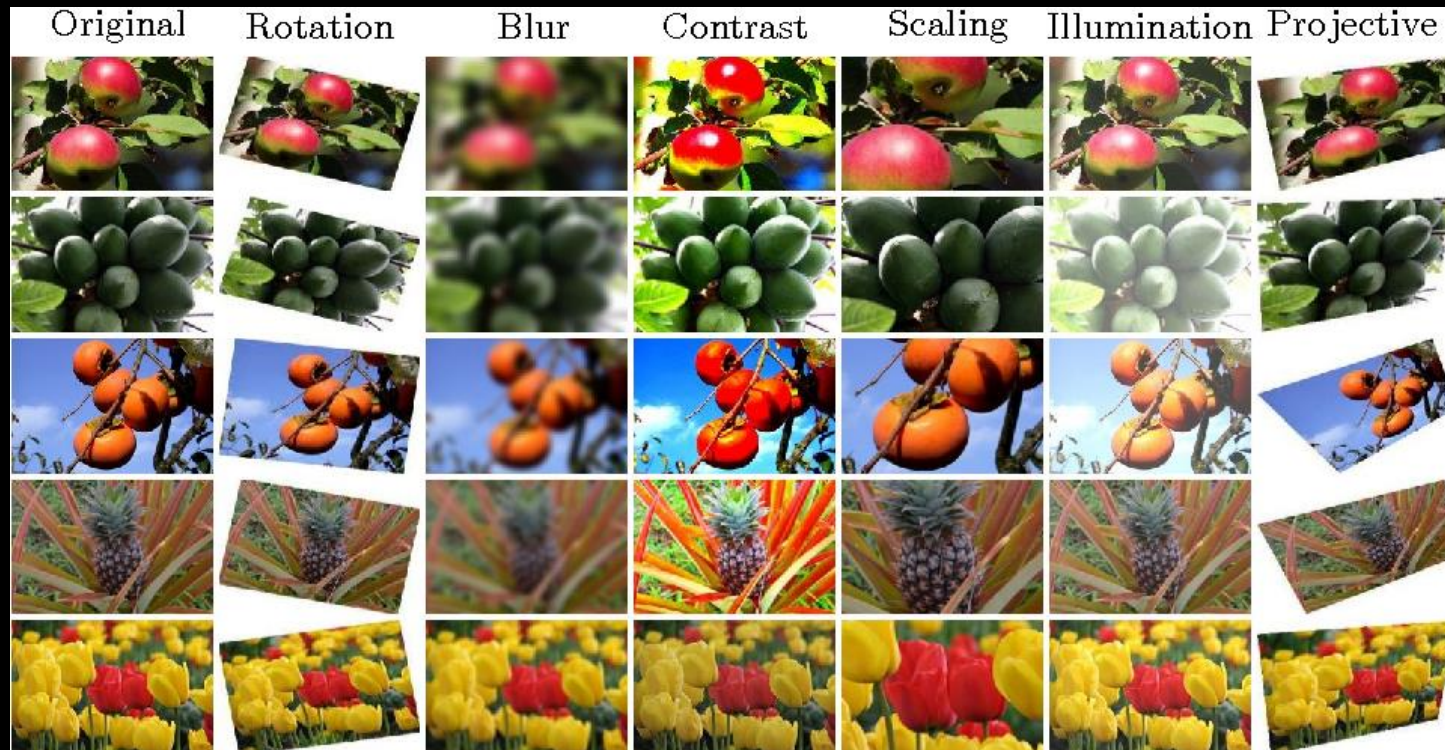
- **Data Augmentation**

- Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data.
- This includes making small changes to data or using deep learning models to generate new data points.



Ways to Clean Data (contd.)

- Data Augmentation



Ways to Clean Data (contd.)

- Data Augmentation
 - Advanced models for data augmentation are
 - **Adversarial training/Adversarial machine learning**: It generates adversarial examples which disrupt a machine learning model and injects them into a dataset to train.
 - **Generative adversarial networks (GANs)**: GAN algorithms can learn patterns from input datasets and automatically create new examples which resemble training data.
 - **Neural style transfer**: Neural style transfer models can blend content image and style image and separate style from content.
 - **Reinforcement learning**: Reinforcement learning models train software agents to attain their goals and make decisions in a virtual environment.

Preprocessing for Text

- Lowering the Text
- Remove Numbers
- Remove URLs
- Remove Symbols
- Remove Punctuations
- Remove Whitespaces
- Remove Stopwords
- Stemming and Lemmatization
- Vectorization and Semantic Representation

Preprocessing for Image

- **Rescaling**
- **Grayscale**
- **Samplewise Centering**
- **Featurewise centering**
- **Augmentation Transformations**

<https://www.intel.com/content/www/us/en/developer/articles/technical/hands-on-ai-part-14-image-data-preprocessing-and-augmentation.html>

Data Formats

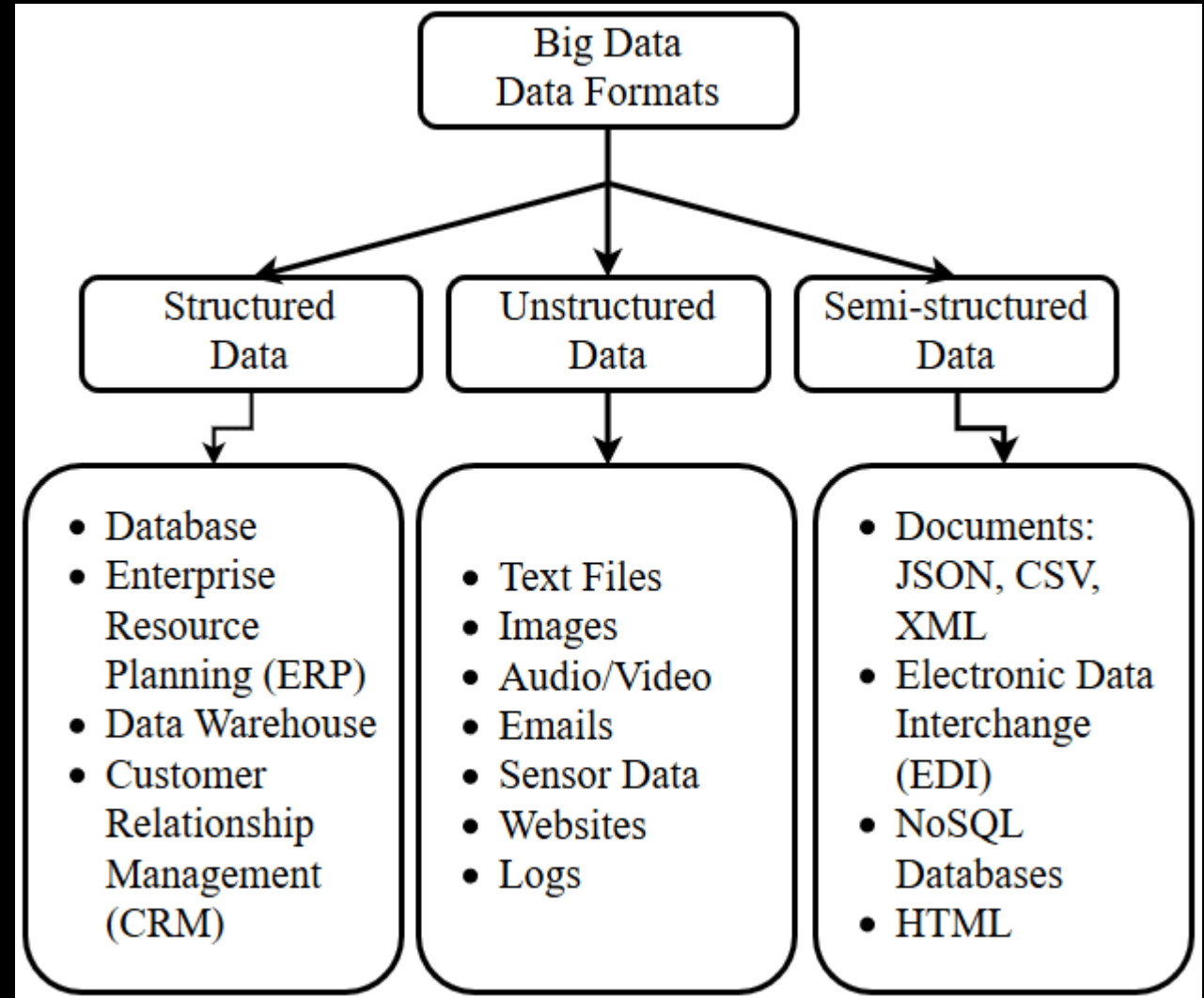
- Data format is the definition of the structure of data within a database or file system that gives the information its meaning.
- How data is organized and presented.
- Arrangement of data fields in a specific way.
- Some information can be organized in different ways. (xml, text, word, pdf).

Why Data Format is important?

- Source data can come in many different data formats.
- To run analytics effectively, a data scientist must first convert that source data to a common format for each model to process.
- With many different data sources and different analytic routines, that data wrangling can take 80 to 90 percent of the time spent on developing a new model.
- Having a model-driven architecture that simplifies the conversion of the source data to a standard, easy-to-use format ready for analytics reduces the overall time required and allows the data scientist to focus on machine learning model development and the training life cycle.

Data Formats

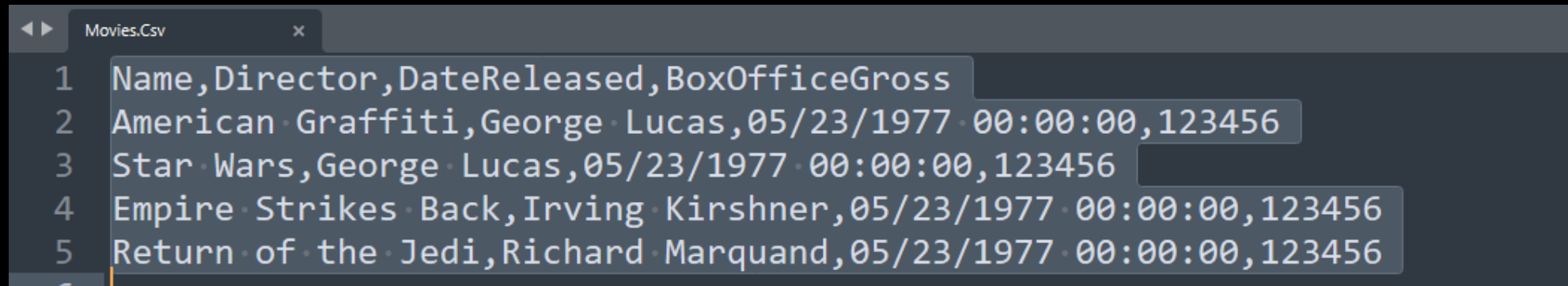
- CSV
- Delimited
- Spreadsheets
- JSON
- XML
- Tabular
- Relational
- Key-Value Pair
- Parquet



Data Formats - CSV

- **CSV (Comma Separated Value)**

- Tabular data but presented in plain text format where each field is separated by comma (,)
- CSV files are often used to transfer data between sources.
- Common format to be manipulated using text editor or commonly Excel.



A screenshot of a text editor window titled 'Movies.Csv'. The editor displays a CSV file with 5 lines of data. The first line is a header row with four columns: 'Name', 'Director', 'DateReleased', and 'BoxOfficeGross'. The subsequent four lines contain movie data, each with a line number in the left margin. The data is as follows:

	Name	Director	DateReleased	BoxOfficeGross
2	American Graffiti	George Lucas	05/23/1977 00:00:00	123456
3	Star Wars	George Lucas	05/23/1977 00:00:00	123456
4	Empire Strikes Back	Irving Kirshner	05/23/1977 00:00:00	123456
5	Return of the Jedi	Richard Marquand	05/23/1977 00:00:00	123456

Data Formats - DSV

- **DSV (Delimited Separated Value)**

- Tabular data but presented in plain text format where each field is separated by different delimiter such as semicolon (;), pipe (|), slash (/), tab , colon (:) etc.
- CSV is also a common and special type of DSV.
- Typically a delimited file format is indicated by a specification.
- Some specifications provide conventions for avoiding delimiter collision, others do not.
- **Delimiter collision** is a problem that occurs when a character that is intended as part of the data gets interpreted as a delimiter instead.

```
Name;RollNumber;Grade  
Aman;21;C  
Rahul;22;A  
Vishal;12;B  
Jyoti;55;A  
Swati;27;B  
Kishan;23;D
```

Data Formats - DSV

- **DSV (Delimited Separated Value)**

- Following are the problems that often occurs with different delimiters
 1. Comma and space separated format often suffer from delimiter Collison.
 2. One problem with tab-delimited text files is that tabs are difficult to distinguish from spaces; therefore, there are sometimes problems with the files being corrupted when people try to edit them by hand.
 3. A delimiter cannot be binary zero, a line-feed character, a carriage-return, or a blank space

Data Formats - Spreadsheets

- **Spreadsheets**

- Spreadsheets are very common computer software for data preparation, presentation, manipulation and analysis.
- We can also import data from various format and convert to wide range of formats.
- Commonly used extensions are .odt, .csv, .xls, .xlsx

Data Formats – JSON

- **JSON (JavaScript Object Notation)**

- JSON is a language-independent open data format that uses human-readable text to express data objects consisting of attribute-value pairs.
- Although originally derived from the JavaScript scripting language, JSON data can be generated and parsed with a wide variety of programming languages.
- JSON is also a plain text data but with a light weighted data interchange format.
- JSON, today, is commonly used with API to exchange data between computers.

```
{ "employees": [  
  { "firstName": "John", "lastName": "Doe" },  
  { "firstName": "Anna", "lastName": "Smith" },  
  { "firstName": "Peter", "lastName": "Jones" }  
]}
```

Data Formats - XML

- XML (Extensible Markup Language)
 - A simple text-based format for representing structured information: documents, data, configuration, books, transactions, invoices, and much more.
 - XML is one of the most widely-used formats for sharing structured information today: between programs, between people, between computers and people, both locally and across networks.
 - XML was designed to store and transport data

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

Data Formats - XML

- XML (Extensible Markup Language)
 - If you are already familiar with HTML, you can see that XML is very similar.
 - However, the syntax rules of XML are strict: XML tools will not process files that contain errors, but instead will give you error messages so that you fix them.
 - This means that almost all XML documents can be processed reliably by computer software.
 - XML Does Not Use Predefined Tags.

Data Formats - Tabular

- **Tabular**

- "Tabular" is simply information presented in the form of a table with rows and columns.
- It represents the structured data format.
- A data table is a neat and convenient way to present a large body of information that includes repeating data elements.
- For example, each entry in a list of company clients contains the client's name, title, address, phone number and other identifying information.
- This information can be listed in tabular format -- that is, in rows and columns -- by using separate columns for each data element.
- Columns are usually identified with headers such as "Client Name," "Street Address" and "Email Address," and each row contains all the information for a single client.

Data Formats - Relational

- **Relational**

- A relational model organizes data into one or more tables (or "relations") of columns and rows, with a unique key identifying each row.
- Rows are also called records or tuples.
- Columns are also called attributes. Generally, each table/relation represents one "entity type" (such as customer or product).
- A system used to maintain relational databases is a **relational database management system (RDBMS)**.
- A relationship is maintained between tables.
- For example, each row of a class table corresponds to a class, and a class corresponds to multiple students, so the relationship between the class table and the student table is "one to many".

Data Formats – Key Value

- **Key-Value format**

- A key-value database, a.k.a key-value store, associates a value (which can be anything from a number or simple string to a complex object) with a key, which is used to keep track of the object.
- In its simplest form, a key-value store is like a dictionary/array/map object as it exists in most programming paradigms, but which is stored in a persistent way and managed by a Database Management System (DBMS).

```
{  
  name: "John",  
  age : 35,  
  dob : ISODate("01-05-1990"),  
  profile_pic : "https://example.com/john.jpg",  
  social : {  
    twitter : "@mongojohn",  
    linkedin : "https://linkedin.com/abcd_mongojohn"  
  }  
}
```

Data Formats – Key Value

- **Key-Value format**

- Key-value databases use compact, efficient index structures to be able to quickly and reliably locate a value by its key, making them ideal for systems that need to be able to find and retrieve data in constant time.
- Redis, for instance, is a key-value database that is optimized for tracking relatively simple data structures (primitive types, lists, heaps, and maps) in a persistent database.

Data Formats – Parquet

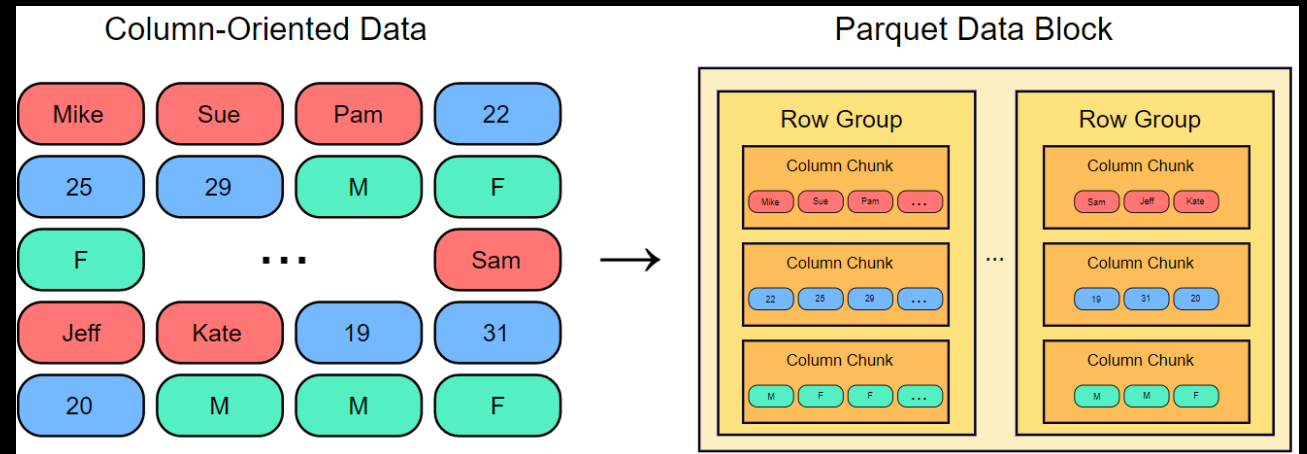
- **Parquet**

- Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval.
- It provides efficient data compression and encoding schemes with enhanced performance to handle complex data in bulk.
- Apache Parquet is designed to be a common interchange format for both batch and interactive workloads.
- It is similar to other columnar-storage file formats available in Hadoop, namely RCFile and ORC.

Data Formats – Parquet

- **Parquet**

- Characteristics of Parquet
- Free and open source file format.
- Language agnostic.
- Column-based format
- Used for analytics (OLAP)
- Highly efficient data compression and decompression.
- Supports complex data types and advanced nested data structures.



Data Formats based on Structure

	<i>Structured</i>	<i>Semi-structured</i>	<i>Unstructured</i>
<i>Examples</i>	<i>Relational databases</i>	<i>Csv, JSON, XML</i>	<i>Word documents, pdfs, web pages.</i>
<i>Flexibility</i>	<i>Low</i>	<i>Medium</i>	<i>High</i>
<i>Robustness</i>	<i>High</i>	<i>Medium</i>	<i>Low</i>
<i>Ease of analysis/query</i>	<i>High</i>	<i>Medium</i>	<i>Low</i>
<i>Human Readability</i>	<i>Low</i>	<i>Medium</i>	<i>High</i>

Row Based vs Column Based



ROW-stores

- "traditional"
- typically OLTP
- partitioned horizontally
- stores all row's values together as a tuple
- write data quickly
- indexing can drastically improve query response time



Columnar-stores

- typically OLAP
- partitioned vertically
- stores multiple row's column values together
- reads are efficient due to only reading relevant columns
- column compression works well, speeding up reads

Wide table format

- Wide, or unstacked data is presented with each different data variable in a separate column.

<i>Person</i>	<i>Age</i>	<i>Weight</i>	<i>Height</i>
<i>Bob</i>	<i>32</i>	<i>168</i>	<i>180</i>
<i>Alice</i>	<i>24</i>	<i>150</i>	<i>175</i>
<i>Steve</i>	<i>64</i>	<i>144</i>	<i>165</i>

Narrow table format

- Narrow, stacked, or long data is presented with one column containing all the values and another column listing the context of the value.

<i>Person</i>	<i>Variable</i>	<i>Value</i>
<i>Bob</i>	<i>Age</i>	<i>32</i>
<i>Bob</i>	<i>Weight</i>	<i>168</i>
<i>Bob</i>	<i>Height</i>	<i>180</i>
<i>Alice</i>	<i>Age</i>	<i>24</i>
<i>Alice</i>	<i>Weight</i>	<i>150</i>
<i>Alice</i>	<i>Height</i>	<i>175</i>
<i>Steve</i>	<i>Age</i>	<i>64</i>
<i>Steve</i>	<i>Weight</i>	<i>144</i>
<i>Steve</i>	<i>Height</i>	<i>165</i>

This is often easier to implement; addition of a new field does not require any changes to the structure of the table, however it can be harder for people to understand.

Why do we need to perform format conversion?

- **System Integration/Compatibility**: to make it easier for systems to talk to each other.
- **Readability**: easier to read and understand.
- **Better Representation**: Some data is better represented in certain formats.
- **Efficiency**: Some data formats are more efficient for analysis (faster/lesser cost).

Data Validation

- Data validation is the practice of checking the integrity, accuracy and structure of data before it is used for a business operation.
- Data validation operation results can provide data used for data analytics, business intelligence or training a machine learning model.
- Data can be examined as part of a validation process in a variety of ways, including data type, constraint, structured, consistency and code validation.
- For example, if data is not in the right format to be consumed by a system, then the data can't be used easily, if at all.

Data Validation

- Each type of data validation is designed to make sure the data meets the requirements to be useful.

Techniques for data validation

1. Data Type Check

- A data type check confirms that the data entered has the correct data type.
- For example, a field might only accept numeric data. If this is the case, then any data containing other characters such as letters or special symbols should be rejected by the system.

<https://www.analyticsvidhya.com/blog/2021/05/data-validation-in-machine-learning-is-imperative-not-optional/>

<https://corporatefinanceinstitute.com/resources/data-science/data-validation/>

Techniques for data validation (contd.)

2. Range Check

- A range check will verify whether input data falls within a predefined range.
- For example, latitude and longitude are commonly used in geographic data. A latitude value should be between -90 and 90, while a longitude value must be between -180 and 180. Any values out of this range are invalid.

Techniques for data validation (contd.)

3. Format Check

- Many data types follow a certain predefined format.
- A common use case is date columns that are stored in a fixed format like “YYYY-MM-DD” or “DD-MM-YYYY.”
- A data validation procedure that ensures dates are in the proper format helps maintain consistency across data and through time.

Techniques for data validation (contd.)

4. Consistency Check

- A consistency check is a type of logical check that confirms the data's been entered in a logically consistent way.
- An example is checking if the delivery date is after the shipping date for a parcel.

END OF THE CHAPTER

Next Chapter: Chapter 4 – Machine Learning