

# Data Pre-Processing-II

(Data Integration, Data Transformation)

---

Dr. JASMEET SINGH  
ASSISTANT PROFESSOR, CSED  
TIET, PATIALA

# Data Integration

---

- **Data Integration:** It is the process of merging the data from multiple sources into a coherent data store.  
e.g. Collection of banking data from different banks at data stores of RBI

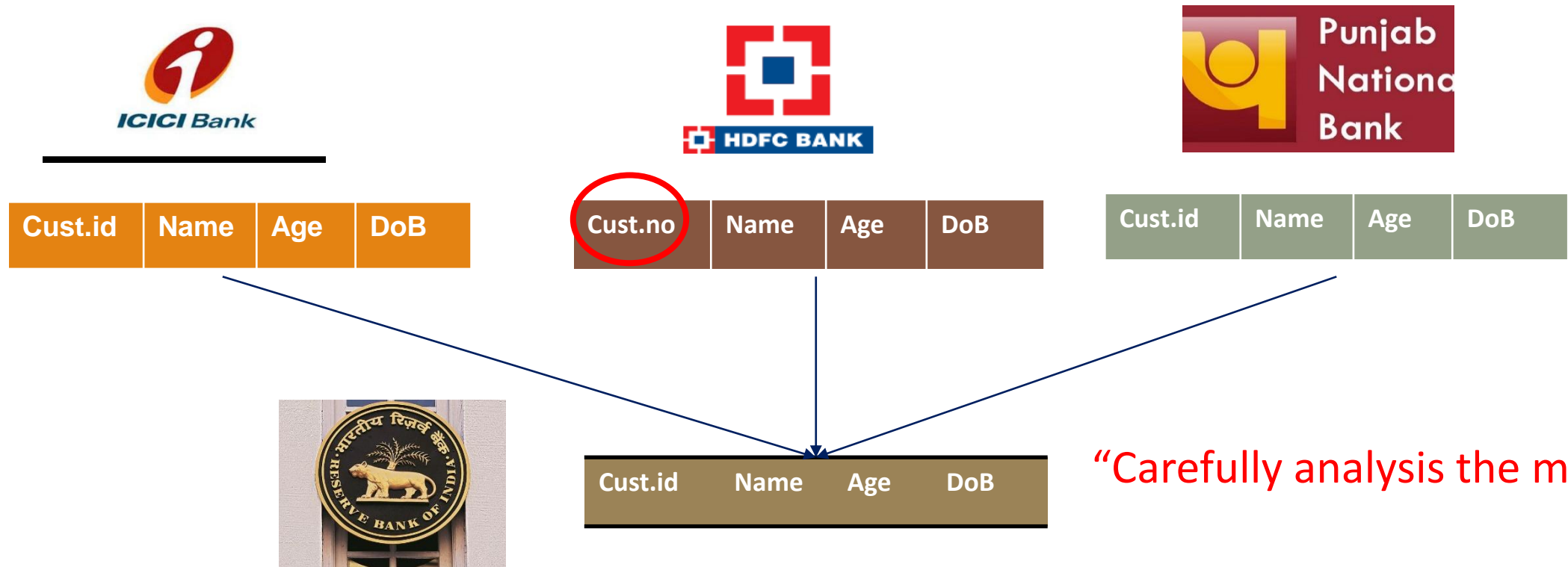
## Issues in data integration

- Schema integration and feature matching
- Detection and resolution of data value conflicts.
- Redundant features



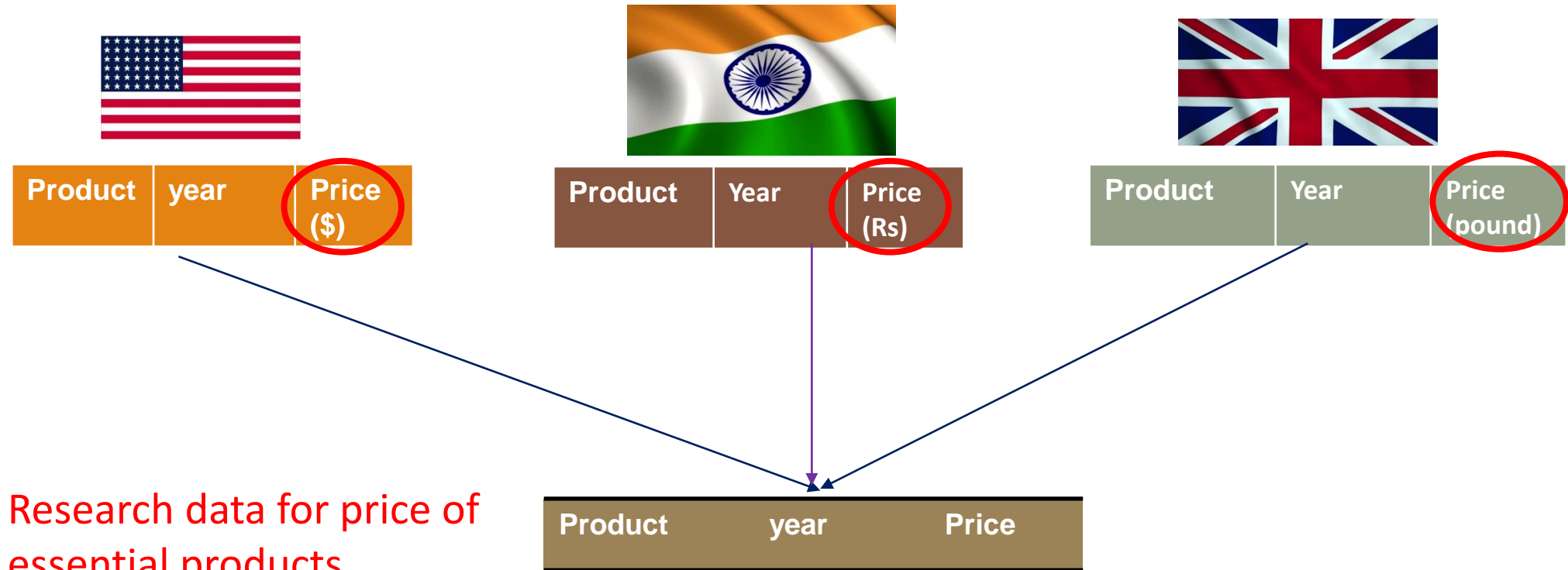
# Data Integration

## Schema integration and feature matching:



# Data Integration

## Detection and resolution of data value conflicts:



“Carefully analysis the metadata”

# Data Integration

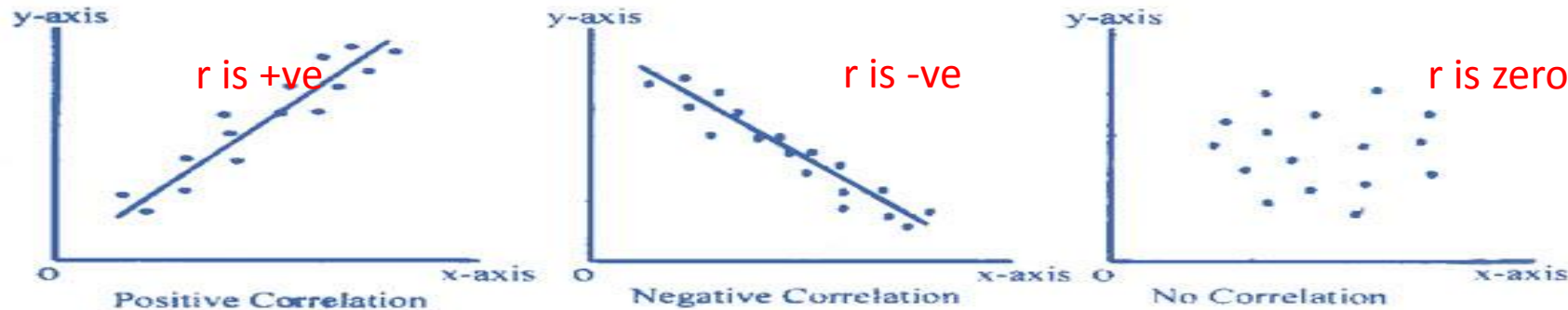
**Redundant features:** They are unwanted features.

- To deal with redundant features correlation analysis is performed. Denoted by  $r$ .
- A threshold is decided to find redundant features.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$



Cust.id	Name	Age	DoB
---------	------	-----	-----



# Data Transformation

---

**Data Transformation:** It is a process to transform or consolidate the data in a form suitable for machine learning algorithms.

**Major techniques of data transformation are :-**

- Normalization/Scaling
- Aggregation
- Generalization
- Feature construction

# Data Transformation- Scaling

---

**Scaling/Normalization:** It is the technique of mapping the numerical feature values of any range into a specific smaller range i.e. 0 to 1 or -1 to 1 etc.

**Popular methods of Normalization are:-**

- Min-Max method
- Mean normalization
- Z score method
- Decimal scaling method
- Log Transformer
- MaxAbs Transformer
- InterQuartile Scaler / Robust Scaler

# Data Transformation- Scaling Contd

## Min-Max method:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where,

$X'$  is mapped value

$X$  is data value to be mapped in specific range

$X_{min}$  and  $X_{max}$  is minimum and maximum value of feature vector corresponding to  $X$ .

x		$X'$
2	Min-max normalization →	0
47		0.512
90		1
18		0.18
5		0.034



# Data Transformation- Scaling Contd

## Mean normalization

$$X' = \frac{X - \bar{X}}{X_{max} - X_{min}}$$

Where,

$X'$  is mapped value

$X$  is data value to be mapped in specific range

$\bar{X}$  is mean of feature vector corresponding to  $X$ .

$X_{min}$  and  $X_{max}$  is minimum and maximum value of feature vector corresponding to  $X$ .

X		X'
2	Mean normalization →	-0.345
47		0.166
90		0.655
18		-0.164
5		-0.311

# Data Transformation- Scaling Contd...

## Z Score method:

$$X' = \frac{X - \bar{X}}{\sigma}$$

Where,

$X'$  is mapped value

$X$  is data value to be mapped in specific range

$\bar{X}$  and  $\sigma$  is mean and standard deviation of feature vector corresponding to  $X$ .

X		X'
2	Z score normalization →	-0.826
47		0.397
90		1.566
18		-0.391
5		-0.745

# Data Transformation- Scaling Contd...

## Decimal scaling method:


$$X' = \frac{X}{10^j}$$

Where,

$X'$  is mapped value

$X$  is data value to be mapped in specific range

$j$  is maximum of the count of digits in minimum and maximum value of feature vector corresponding to  $X$

X		X'
2	Decimal scaling normalization 	0.02
47		0.47
90		0.9
18		0.18
5		0.05

# Data Transformation- Scaling Contd...

## Log Transformer:

$$X' = \log(X)$$

Where,

$X'$  is mapped value

$X$  is data value to be mapped in specific range

It is primarily used to convert a skewed distribution to a normal distribution/less-skewed distribution.

The log operation had a dual role:

- Reducing the impact of too-low values
- Reducing the impact of too-high values.

X		X'
2	Log scaling normalization →	0.693147
47		3.850148
90		4.499810
18		2.890372
5		1.609438

# Data Transformation – Scaling Contd...

## MaxAbs Scaler

It first takes the absolute value of each value in the column and then takes the maximum value out of those.

This operation scales the data between the range  $[-1, 1]$ .

X		X'
100	MaxAbs normalization →	0.05
-263		-0.1315
-2000		-1
18		0.009
5		0.0025

# Data Transformation- Scaling Contd

## Interquartile/Robust normalization

$$X' = \frac{X - Q1}{Q3 - Q1}$$

Where,

$X'$  is mapped value

$X$  is data value to be mapped in specific range

- The mean, maximum and minimum values of the columns. All these values are sensitive to outliers.
- If there are too many outliers in the data, they will influence the mean and the max value or the min value.
- Thus, even if we scale this data using the above methods, we cannot guarantee a balanced data with a normal distribution.

x		$X'$
2	Robust normalization →	-0.38095238
47		0.69047619
90		1.71428571
18		0
5		-0.30952381

# Data Transformation- Aggregation

---

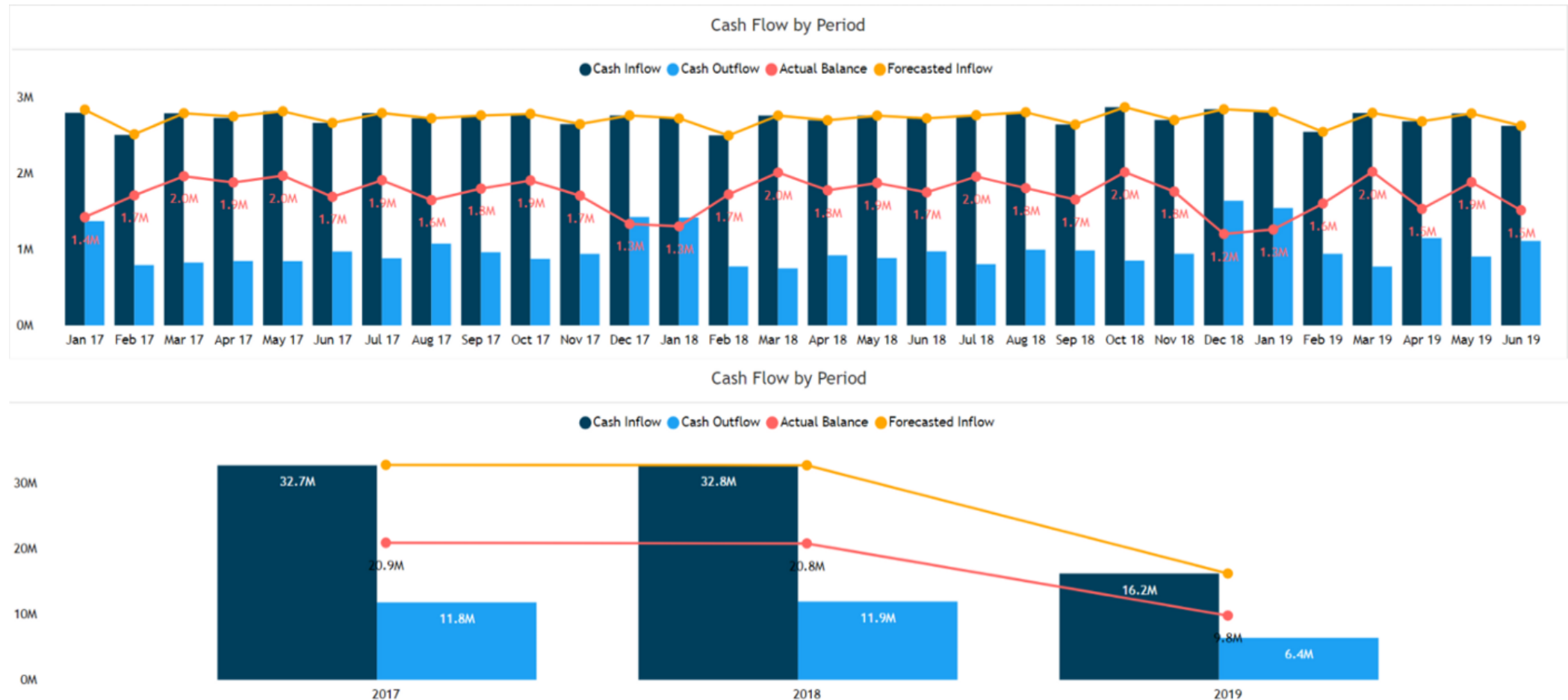
**Aggregation** : take the aggregated values in order to put the data in a better perspective.

e.g. in case of transactional data, the day to day sales of product at various stores location can be aggregated store wise over months or years in order to be analyzed for some decision making.

## **Benefits of aggregation**

- Reduce memory consumption to store large data records.
- Provides more stable view point than individual data objects

# Data Transformation- Aggregation Contd..





# Data Transformation- Generalization

---

**Generalization:** The data is generalized from low-level to higher order concepts using concept hierarchies.

e.g. categorical attributes like street can be generalized to higher order concepts like city or country.

“The decision of generalization level depends on the problem statement”

# Data Transformation- Feature Construction

---

- Feature construction involves transforming a given set of input features to generate a new set of powerful features.
- For e.g. feature like mobile number and landline number combined together under new feature contact number.
- Features like apartment length and breadth must be converted to apartment area.

# Data Transformation- Feature Construction

---

- There are certain situations where feature construction is an essential activity before we can train a machine learning model.
- These situations are:
  - When features have categorical value and machine learning needs numeric value inputs.
    - ✓ Label Encoding
    - ✓ One-Hot Encoding
    - ✓ Dummy Encoding
  - When features have numeric (continuous) value and need to be converted to categorical values.
    - Rank according to numerical continuous values
    - Use Evaluation metrics like Gain Ratio, Gini Index to convert a numerical feature into binary categorical.
  - When text-specific feature construction need to be done.
    - Bag-of-words
    - Tf-idf
    - Word Embeddings

# Data Transformation- Feature Construction

## ➤ Label Encoding

This approach is very simple and it involves converting each value in a column to a number.

BRIDGE-TYPE (TEXT)	BRIDGE-TYPE (NUMERICAL)
Arch	0
Beam	1
Truss	2
Cantilever	3
Tied Arch	4
Suspension	5
Cable	6

- Depending upon the data values and type of data, label encoding induces a new problem since it uses number sequencing.
- The problem using the number is that they introduce relation/comparison between them.
- The algorithm might misunderstand that data has some kind of hierarchy/order  $0 < 1 < 2 \dots < 6$  and might give 6X more weight to 'Cable' in calculation than 'Arch' bridge type

# Data Transformation- Feature Construction

## ➤ Label Encoding

- Let's consider another column named 'Safety Level'.
- Performing label encoding of this column also induces order/precedence in number, but in the right way.
- Here the numerical order does not look out-of-box and it makes sense if the algorithm interprets safety order  $0 < 1 < 2 < 3 < 4$  i.e. none < low < medium < high < very high.

SAFETY-LEVEL (TEXT)	SAFETY-LEVEL (NUMERICAL)
None	0
Low	1
Medium	2
High	3
Very-High	4

# Data Transformation- Feature Construction

---

## ➤ One-Hot Encoding

- Though label encoding is straight but it has the disadvantage that the numeric values can be misinterpreted by algorithms.
- The ordering issue is addressed in another common alternative approach called ‘One-Hot Encoding’.
- In this strategy, each category value is converted into a new column and assigned a 1 or 0 (notation for true/false) value to the column.
- It does have the downside of adding more columns to the data set.
- It can cause the number of columns to expand greatly if you have many unique values in a category column.

# Data Transformation- Feature Construction

## ➤ One-Hot Encoding

BRIDGE-TYPE (TEXT)	BRIDGE-TYPE (Arch)	BRIDGE-TYPE (Beam)	BRIDGE-TYPE (Truss)	BRIDGE-TYPE (Cantilever)	BRIDGE-TYPE (Tied Arch)	BRIDGE-TYPE (Suspension)	BRIDGE-TYPE (Cable)
Arch	1	0	0	0	0	0	0
Beam	0	1	0	0	0	0	0
Truss	0	0	1	0	0	0	0
Cantilever	0	0	0	1	0	0	0
Tied Arch	0	0	0	0	1	0	0
Suspension	0	0	0	0	0	1	0
Cable	0	0	0	0	0	0	1

# Data Transformation- Feature Construction

---

## ➤ Dummy Encoding

- Dummy coding scheme is similar to one-hot encoding.
- This categorical data encoding method transforms the categorical variable into a set of binary variables (also known as dummy variables).
- In the case of one-hot encoding, for  $N$  categories in a variable, it uses  $N$  binary variables.
- The dummy encoding is a small improvement over one-hot-encoding. Dummy encoding uses  $N-1$  features to represent  $N$  labels/categories.



# Data Transformation- Feature Construction

## ➤ Dummy Encoding

BRIDGE-TYPE (TEXT)	BRIDGE-TYPE (Arch)	BRIDGE-TYPE (Beam)	BRIDGE-TYPE (Truss)	BRIDGE-TYPE (Cantilever)	BRIDGE-TYPE (Tied Arch)	BRIDGE-TYPE (Suspension)
Arch	1	0	0	0	0	0
Beam	0	1	0	0	0	0
Truss	0	0	1	0	0	0
Cantilever	0	0	0	1	0	0
Tied Arch	0	0	0	0	1	0
Suspension	0	0	0	0	0	1
Cable	0	0	0	0	0	0

# Data Transformation- Feature Construction

## Text specific Features- BoW , TF-IDF

Document A: The Car Is Driven On The Road

Document B: The Truck is Driven on the highway

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	$\log(2/2) = 0$	0	0
Car	1/7	0	$\log(2/1) = 0.3$	0.043	0
Truck	0	1/7	$\log(2/1) = 0.3$	0	0.043
Is	1/7	1/7	$\log(2/2) = 0$	0	0
Driven	1/7	1/7	$\log(2/2) = 0$	0	0
On	1/7	1/7	$\log(2/2) = 0$	0	0
The	1/7	1/7	$\log(2/2) = 0$	0	0
Road	1/7	0	$\log(2/1) = 0.3$	0.043	0
Highway	0	1/7	$\log(2/1) = 0.3$	0	0.043

# TF-IDF formula detail

---

**Term Frequency:** TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

**Inverse Document Frequency:** IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and).

$$IDF = \log\left(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}}\right)$$

The TF-IDF of a term is calculated by multiplying TF and IDF scores.

$$TF\text{-}IDF = TF \times IDF$$