

# Data Preprocessing

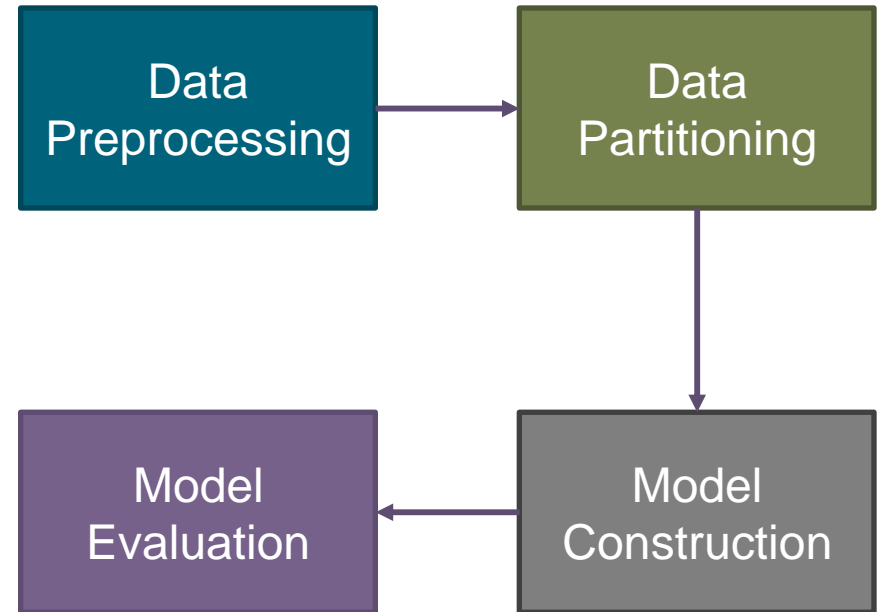
---

## Data Cleaning: Part I

# The Data Science Workflow

---

- Although every data science project is different, the steps involved in the data science process can be formalized.
- **The data science process consist of data preprocessing, data partitioning, model construction, and model evaluation.**



# Data Preprocessing

---

- Data preprocessing involves preparing the data and making it ready for modelling.
- Data preprocessing involves:
  - Data cleaning
  - Transformation
  - Feature extraction
  - Data exploration

Preprocessing Task	Activities
Data cleaning	Remove noisy data, fix inconsistent data, handle missing data and outliers.
Data transformation	Scale the data to fall within an appropriate range such as 0 to 1 (normalization), transform the data to an appropriate format, etc.
Feature extraction	Data reduction, reduce data size by eliminating redundant or meaningless features
Data exploration	Descriptive statistics and visualization

# Data Cleaning

---

- Data cleaning is applied to increase the data quality.
- Three key components of **data quality** include:
  - accuracy,
  - consistency,
  - completeness.
- **Inaccurate or noisy** data contains errors or values that deviate from the expected.
- **Inconsistent data** is data with inconsistent naming convention, for example date formats.
- **Incomplete data** has missing values or attributes of interest.

# Data Cleaning (cont.)

## Components of data cleaning:

- Handle missing data
- Removing outliers and smoothen noisy data
- Fix inconsistencies

	Credit Score	Missed Payments	Home Owner	Credit Age	Time On Job	Loan Status
1	600	2	0	5.6	7	1
2	649	0	1	5.0	7	1
3	485	4	0	7.4	6	0
4	624	2	1	7.4	6	0
5	650	15	0		7	5

Outlier

Missing value

Incorrect entry. Loan status is binary with possible values of {0, 1}.

# Data Cleaning: Handling Missing Data

---

## Identify missing data

- To handle missing data, first identify if there are missing values in the data.
- Note that missing values can take different forms including:
  - Blanks,
  - N/A, NA, NAN
  - 999, -1, etc., as in code book
- Check unique values for each attribute. This can be helpful in identifying missing values.
- Check the number of missing values per attribute.
  - In Python, you can use the code:  
`DataFrame.isnull().sum()`

# Data Cleaning:

## Handling Missing Data (cont.)

---

### **Missing data mechanism:**

- How missing data is handled depends on the mechanism of the missing data.
  - The three mechanisms or patterns of missing data are:
    - MCAR
    - MAR
    - NMAR
- **MCAR:** Missing Completely at Random
  - **MAR:** Missing at Random
  - **NMAR:** Not Missing at Random
  - The pattern of missingness is based on how the missing data relates with the variables.

# Data Cleaning:

## Handling Missing Data (cont.)

---

### **MCAR:**

- Data is missing completely at random when missing data on a specific variable is unrelated to the data on the other variables.
- The variables in the data are not responsible for the missing data.

### **MAR:**

- This is when the missing values are related with some other variable(s).
- Some variable is responsible for the missing data. For example, male participants may be missing data on their depression level.



# Data Cleaning:

## Handling Missing Data (cont.)

---

### **NMAR:**

- The missing data on a specific variable is related to the values of the variable itself and to other variables
- The variable on which data is missing is responsible for the missing values, and other variables.
- For example, people with expensive houses may have missing data on taxes (high taxes).
- So missing tax values are for high taxes and this is related with expensive houses too.

# Data Cleaning:

## Handling Missing Data (cont.)

---

### **Statistical test of MCAR:**

- To test whether missing data on a variable is MCAR, recode the values of that variable into 1 (=missing) and 0 (=not missing). Then, use other continuous variables to run a series of t-tests.
- If the tests are non-significant, that means the distributions of the missing and non missing data on the other variables are the same.
- Then, we can conclude that the missing data is MCAR, otherwise it is MAR or NMAR.

# Data Cleaning:

## Handling Missing Data (cont.)

---

### **Delete or drop rows:**

- If less than 5% of cases have missing data and
- Missing data mechanism is MCAR or MAR

### **Mean, median, mode imputation:**

- If more than 5% of cases have missing data and missing data mechanism is MCAR only.

### **Use a regression model to predict missing values:**

- If more than 5% of cases have missing data and missing data mechanism is MAR

### **Other imputation techniques:**

- EM algorithm, maximum likelihood, and multiple imputation are better for data MCAR and MNAR.

# Data Cleaning:

## Handling Missing Data (cont.)

---

- **Drop a variable**
  - If 90% or more, of the attribute values are missing and pattern of missingness is MCAR.
- **Caution**
  - How you handle missing data is fairly subjective but be cautious so as not to introduce bias through imputation.
- **Drop and fill syntax**
  - The pandas DataFrame has the **.dropna()** method for dropping rows or columns in Python.
  - The pandas DataFrame also has the **.fillna()** method for single value imputation.

Data Preprocessing: Data Cleaning Part I

---

# The End

# Data Preprocessing

---

## Data Cleaning: Part II

# Data Cleaning: Inaccurate Data

---

## **Sources of inaccurate data**

- Typographic errors can lead to incorrect values.
- Misspelled categorical values can create extra value for the variable.
- Measurement or typographical errors can result to outliers.
- Duplicate data is another source of error as repetition of observation causes that observation to have more influence on the results.
- So, always check the data file and address any errors found in the dataset.

# Data Cleaning: Outliers

---

- Outliers are unusual and inconsistent data values.
- These are values that deviate remarkably from the expected values.
- Outliers could be valid values or could be inaccurate or erroneous data.
- Some inferential statistical techniques are sensitive to outliers, leading to invalid results.
- Algorithms that depend on distance metrics can also produce invalid results when outliers are used.



# Data Cleaning: Outliers (cont.)

---

- Numerical and graphical methods are commonly used to identify univariate outliers.
- Numerical methods
  - Z-score method
  - Inter quartile range method
- Graphical methods:
  - Histogram
  - Scatter plot
  - Boxplot

# Data Cleaning: Outliers (cont.)

---

## Numerical methods:

### Z-Score

- Transform numerical data to z-scores.
- Filter z-score above 3 as outlier.

### Inter quartile range

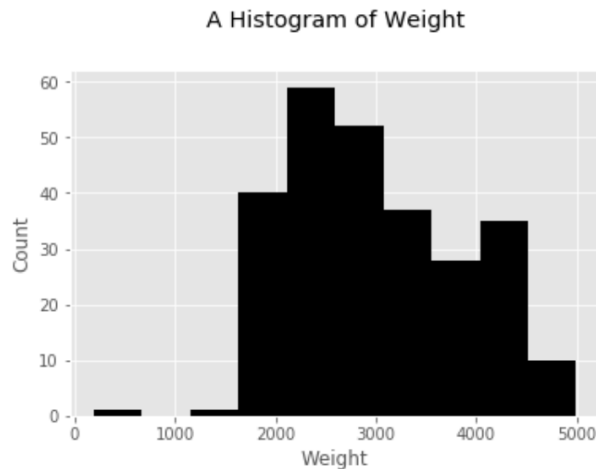
- Find first quartile Q1 (25<sup>th</sup> percentile).

- Find third quartile Q3 (75<sup>th</sup> percentile).
- Compute the interquartile range:  
 $IQR = Q3 - Q1$ .
- Filter values below  $Q1 - 1.5 * IQR$  and values above  $Q3 + 1.5 * IQR$  as outliers.

# Data Cleaning: Outliers (cont.)

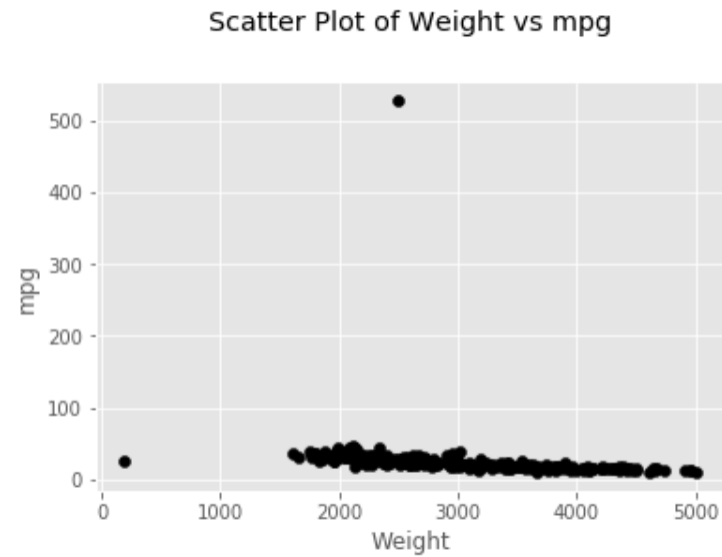
---

## Graphical method: histogram



## Graphical method: scatter plot

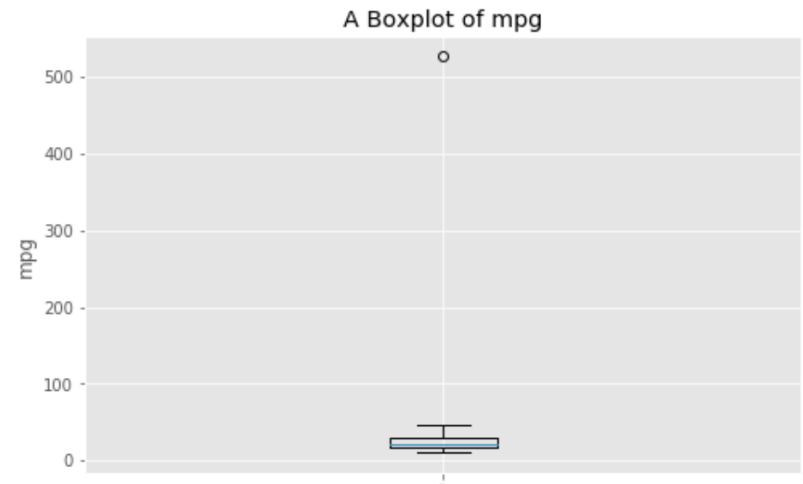
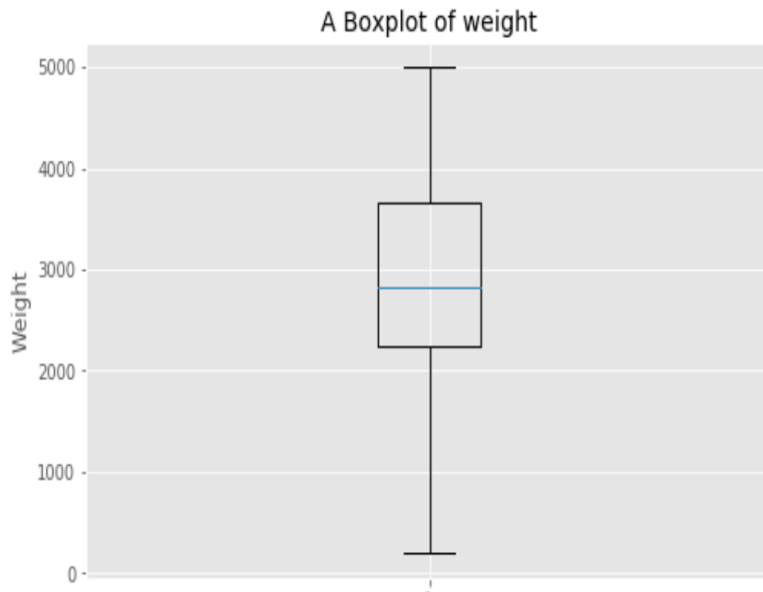
- Scatter plot



# Data Cleaning: Outliers (cont.)

---

## Graphical method: boxplot



# Data Cleaning: Outliers (cont.)

---

## Handling outlier

- Note that outliers are usually handled by removing the outlier.
- The effect of outliers can also be minimized by increasing the sample size.

# Data Exploration

---

- Data exploration is an attempt to understand the data using descriptive statistics and visual plots or graphs.
- Data exploration can enable us to understand the structure and the distribution of the data, relationships between variables, extreme values.
- Descriptive statistics commonly used included mean, mode, median, maximum value, minimum value, variance, standard deviation, skewness and kurtosis.
- Correlations are also used to explore relationships.

# Data Exploration (cont.)

---

- The visualization part of data exploration usually includes histograms, bar charts, boxplots, scatter plots or scatter matrix.
- So, descriptive statistics and visualization are an important part of data preprocessing.



Data Preprocessing: Data Cleaning Part II

---

# The End



# Data Transformation

---

# Data Transformation

---

- In data transformation, the data is consolidated into forms appropriate for the learning algorithm.
- Data transformation can improve the efficiency of the learning algorithm.
- Data can be transformed by:
  - Scaling (also called feature scaling or normalization)
  - Discretization (binning)
  - Using numerical values to represent categorical values

# Data Transformation (cont.)

---

## Scaling

- Scaling or normalization involves transforming attribute values to fall within a smaller range such as -1 to 1, 0 to 1, etc.
- Scaling makes attribute values comparable.
- Normalization is very important especially if the learning algorithm computes distances between data points.
- Normalization ensures that attributes with larger values do not dominate the distance results.

# Data Transformation (cont.)

---

## Min-max normalization

- Min-max normalization finds how far each value is from the minimum value, then scales the difference by the range.

$$x_{scaled} = \frac{x - \min(X)}{\max(X) - \min(X)}$$

- Min-max normalization rescales the data set such that all feature values are in the range  $[0, 1]$ .
- Min-max normalization is sensitive to outliers.

# Data Transformation (cont.)

---

## Z-score standardization

- The z-score standardization finds the difference between each attribute value and the mean of the attribute, then scales the difference by the standard deviation of that attribute.

$$z_{score} = \frac{x - \text{mean}(X)}{\text{standard\_deviation}(X)}$$

- Z-score standardization is sensitive to outliers.
- An attribute with z-scores has a mean of 0 and a standard deviation of 1.

# Data Transformation (cont.)

---

## Decimal scaling

- Decimal scaling transforms all values of an attribute to fall between -1 and 1.

$$x_{decimal} = \frac{x}{10^d}$$

- $d$  is the number of digits in the data value with the largest absolute value.
- For example, if the largest absolute value is  $||120|| = 120$ , then  $d = 3$ .

# Data Transformation (cont.)

---

## Achieve normality

- Some algorithms or statistical methods require that the data or attribute values be normally distributed.
- Attribute values with skewness between -1 and 1 are approximately normal.

$$Skewness = \frac{3 * (\text{mean} - \text{median})}{\text{standard deviation}}$$

- Log transformation, square root transformation and inverse square root transformation are commonly used to eliminate skewness or to achieve normality.

# Data Transformation (cont.)

---

## **Discretization (binning).**

- Discretization is the transformation of numerical attributes into a categorical attributes.
- Discretization is achieved by partitioning the numerical attribute values into a finite set of bins.
- Different types of discretization include:
  - Equal width binning—creates bins of equal width.
  - Equal frequency binning—creates bins with approximately equal number of data points.
  - Binning by clustering—clusters are used as bins.



# Data Transformation (cont.)

---

## Categorical to numerical attributes

- Categorical values that can be ordered could be transformed into numerical values.
- For example, ratings such as **bad**, **good**, and **excellent** can be transformed into 1, 2 and 3 respectively.




ID	Cat_Rating	Num_Rating
101	Good	2
102	Good	2
103	Bad	1
104	Excellent	3
105	Good	2
106	Bad	1

# Data Transformation (cont.)

## Categorical to numerical attributes

- Avoid converting categorical values into numerical values when there is no order, because some algorithms can consider the numbers to be meaningful.
- Instead create dummy variables or use one-hot encoding

Avoid



ID	Cat_Color	Num_Cat
101	red	1
102	green	2
103	blue	3
104	blue	3
105	red	1
106	green	3

# Data Transformation (cont.)

## Full dummy variables

- Full dummy variables are one-hot encodings where the attribute values become dummy variables.
- Dummy variables have binary values of 0 and 1 indicating the absence or presence of that value for each case.

R, G, and B are dummy variables

ID	Cat_Color
101	red
102	green
103	blue
104	blue
105	red
106	green

ID	R	G	B
101	1	0	0
102	0	1	0
103	0	0	1
104	0	0	1
105	1	0	0
106	0	1	0

# Data Transformation (cont.)

---

## Dummy variables with reference group

- K-1 dummy variables are created where k is the number of unique values of the attribute.
- For example, red and green are used as dummy variables while blue is a reference.

ID	Cat_Color	ID	R	G
101	red	101	1	0
102	green	102	0	1
103	blue	103	0	0
104	blue	104	0	0
105	red	105	1	0
106	green	106	0	1

Data Transformation

---

# The End

# Feature Selection and Extraction

---

# Feature Selection and Extraction

---

- Feature selection and extraction are two approaches to dimensionality reduction.
- **Feature selection** focuses on selecting a meaningful subset of existing features from the dataset.
- **Feature extraction** focuses on creating a new set of lower dimensional features from the original features such that the information in the original data is preserved.

# Feature Selection and Extraction (cont.)

---

## **Why is dimensionality reduction necessary?**

- Too many features can increase the run time of the learning algorithm.
- Dimensionality reduction is necessary to solve the problem of the curse of dimensionality.
- The curse of dimensionality is that, the greater the number of features, the greater the number of examples needed to train the algorithm.
- Too many features can lead to overfitting, which can be solved through dimensionality reduction.



# Feature Selection and Extraction (cont.)

---

- In feature selection, the most informative feature are selected.
- For example, the ID3 algorithm, used for constructing decision trees, finds the most informative feature during each iteration.
- Feature extraction involves dimensionality reduction using techniques such as:
  - Principal Component Analysis (PCA)
  - Latent Discriminant Analysis (LDA)
- Feature extraction is usually a linear transformation.

# Feature Selection and Extraction (cont.)

---

- So, feature selection and extraction enable learning algorithms to run faster
- Dimensionality reduction also helps solve the problem of overfitting that results from having too many attributes.
- Feature extraction and selection can be used in text processing, to find the words that best represent a document.
- Step-wise forward selection and backward selection are used in regression to find the best features.

# Sampling

---

- Sampling can be used as a data reduction technique.
- Sampling involves randomly selecting a representative subset of the data and using the subset instead of the entire dataset.
- Sampling is useful if the entire data set is very large, which can result to poor run time.
- Running the rest of the analysis using the subset can be more efficient in terms of time and space.

Feature Selection and Extraction

---

# The End