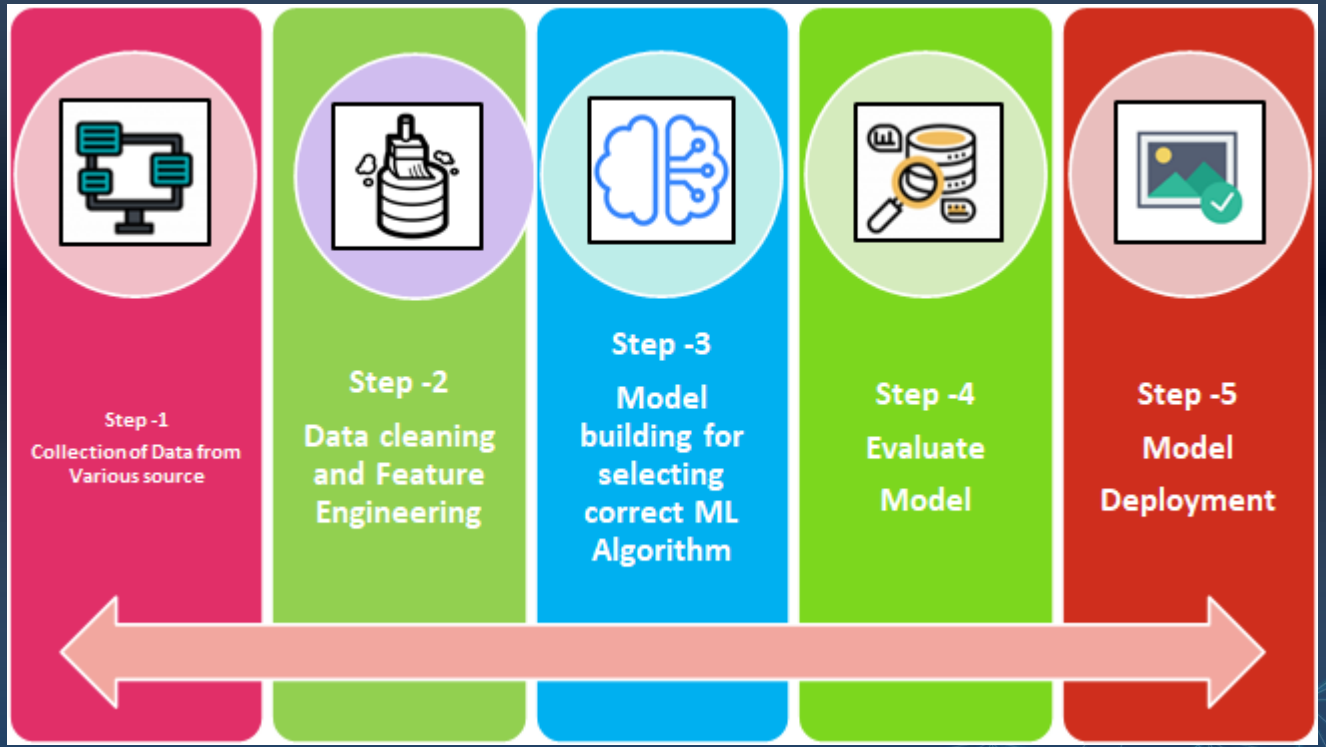
Data Analysis & Preprocessing

Eng- Mohamed Khaled Idris
Eng- Mayar Swilam

Review:


Step -1
Collection of Data from Various source
Step -2
Data cleaning and Feature Engineering
Step -3
Model building for selecting correct ML Algorithm
Step -4
Evaluate Model
Step -5
Model Deployment

1. **Ask**: Business Challenge/Objective/Question
2. **Prepare**: Data generation, collection, storage, and data management
3. **Process**: Data cleaning/data integrity
4. **Analyze**: Data exploration, visualization, and analysis
5. **Share**: Communicating and interpreting results
6. **Act**: Putting your insights to work to solve the problem

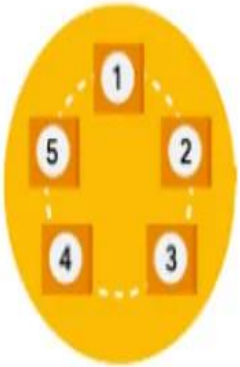
What is Data Analysis ? (Google Data Analytics)

Ask

Ask questions and define the problem.

Prepare

Prepare data by collecting and storing the information.



Process

Process data by cleaning and checking the information.

Analyze

Analyze data to find patterns, relationships, and trends.

Share

Share data with your audience.

Act

Act on the data and use the analysis results.

In other words (UDACITY)

1. Question
2. Wrangle
3. Explore
4. Draw Conclusions
5. Communicate

Step 1: Ask Questions

- Given data then ask questions, or
- Ask questions then **gather** data

Step 2: Wrangle Data

a. **Gather** data to answer question

b. **Assess** data to identify any problems in your data's quality or structure

c. **Clean** data by modifying, replacing, or removing data

Step 3: Perform Exploratory Data Analysis (EDA)

- **Explore then augment** data to maximize the potential of
  - analyses & visualizations & models
- **Exploring** involves:
  - finding **patterns** in data
  - **visualizing** relationships in data
  - building **intuition** about what you're working with
- **After Exploring** (**optional**)
  - **Remove Outliers:**
  - **Feature Engineering:** create better features from data

Step 4: Draw Conclusions (or even make predictions)

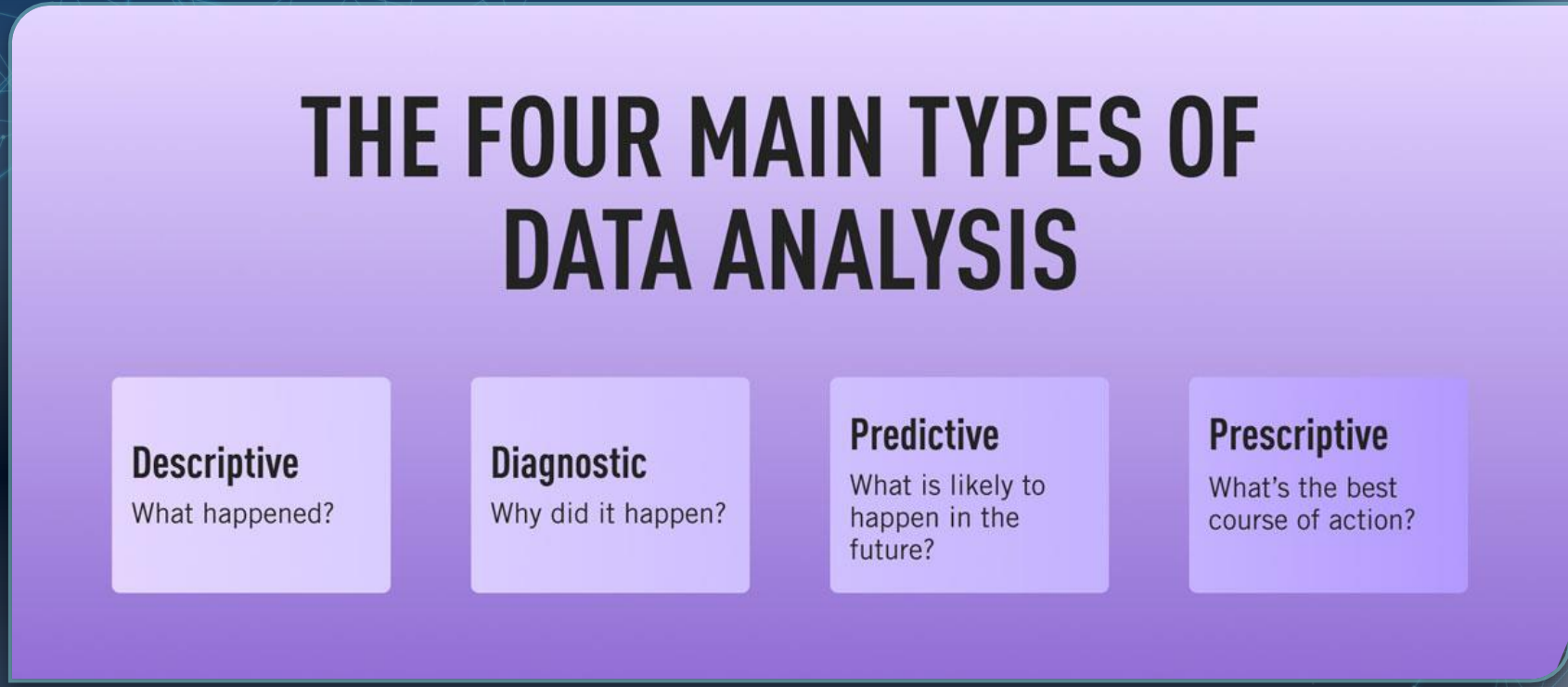
- typically approached with **ML** or **inferential statistics**

Step 5: Communicate Results

- often need to **justify** and **convey** meaning in the insights
- if your end goal is to build a system, you usually need to:
  - **share** what you've built
  - **explain** how you reached design decisions
  - **report** how well it performs
- communicate results by: report | slides | presentation | post | email | conversation
- **Data Visualization** will always be very valuable

Data + business knowledge = mystery solved

Blending data with business knowledge, plus maybe a touch of gut instinct, will be a common part of your process as a junior data analyst. The key is figuring out the exact mix for each particular project. A lot of times, it will depend on the goals of your analysis. That is why analysts often ask, “How do I define success for this project?”



Types of Analysis

That brings us into “Data Quality”

What is the dimensions of “Data Quality” ?

- Data Quality is a measurement of the degree to which data is fit for purpose. Good data quality generates trust in data. Data Quality Dimensions are a measurement of a specific attribute of a data's quality.

Completeness

Completeness measures the degree to which all expected records in a dataset are present. At a data element level, completeness is the degree to which all records have data populated when expected.

Completeness

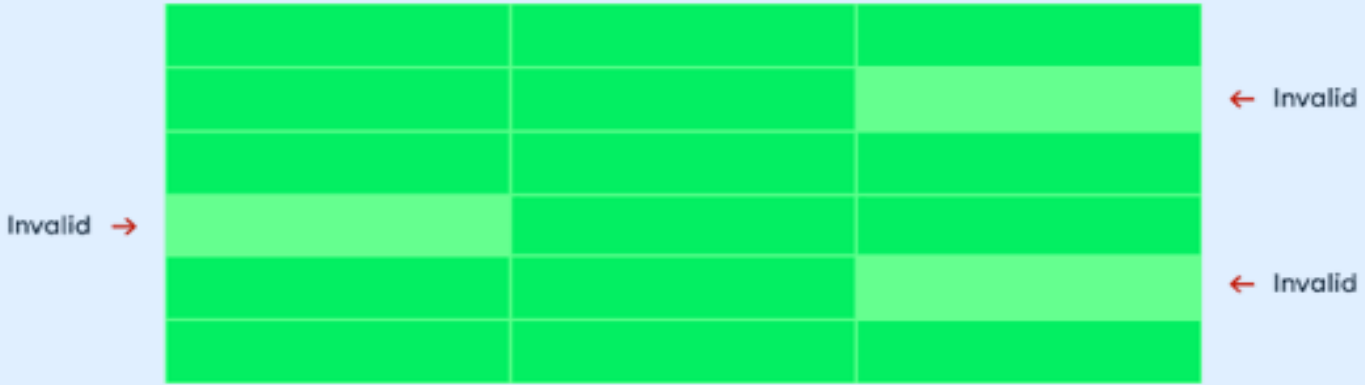
CustomerID	CustomerName	CustomerBirthDate	CustomerAccountType	CustomerAccountBalance	LatestAccountOpenDate
100000192	Robert Brown	4/12/2000	Loan	40390.00	12/20/2026
100000198	Maria Irving	12/1/2025	Deposit	-13280.00	10/21/2018
100000120	Ava Shiffer	10/31/1990	Credit Card	320	3/1/2020
100000192	Robert Brown	4/12/2000	Deposit	40390.00	12/20/2026
100000124	Matthew Martin	5/9/1965	Deposit	70102.00	5/4/2022
100000149		2/4/1988	Loan	0.00	9/20/1990

All records must have a value populated in the CustomerName field.

Example

- **Validity**

Validity measures the degree to which the values in a data element are valid.



Validity

- **Uniqueness**

Uniqueness measures the degree to which the records in a dataset are not duplicated.

Uniqueness

All records must have a unique CustomerID and CustomerName.

CustomerID	CustomerName	CustomerBirthDate	CustomerAccountType	CustomerAccountBalance	LatestAccountOpenDate
100000192	Robert Brown	4/12/2000	Loan	40390.00	12/20/2026
100000198	Maria Irving	12/1/2025	Deposit	-13280.00	10/21/2018
100000120	Ava Shiffer	10/31/1990	Credit Card	320	3/1/2020
100000192	Robert Brown	4/12/2000	Deposit	40390.00	12/20/2026
100000124	Matthew Martin	5/9/1965	Deposit	70102.00	5/4/2022
100000149		2/4/1988	Loan	0.00	9/20/1990

Example

SLA	Table Load Time
08:00 am	07:59 am
10:00 am	09:59 am
11:00 am	11:01 am

Timeliness

Timeliness is the degree to which a dataset is available when expected and depends on service level agreements being set up between technical and business resources.

← Missed the SLA

Timeliness

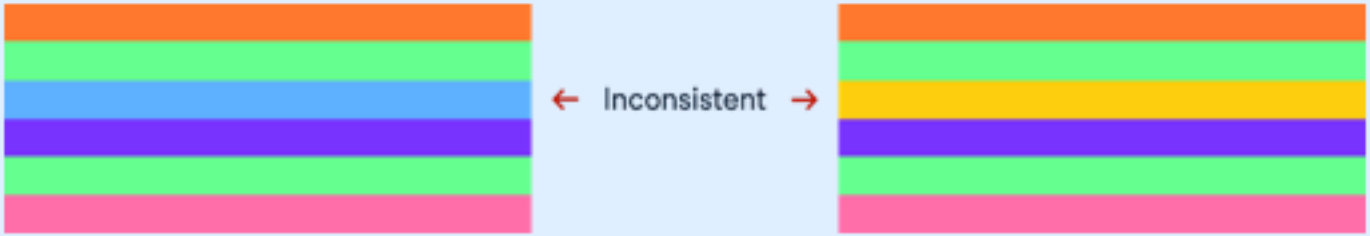
All records in the customer dataset must be loaded by the 9:00 am.

CustomerID	CustomerName
100000192	01-01-2023 11:07 am
100000198	01-01-2023 11:07 am
100000120	01-01-2023 11:07 am

Example

- **Consistency**

Consistency is a data quality dimension that measures the degree to which data is the same across all instances of the data. Consistency can be measured by setting a threshold for how much difference there can be between two datasets.



Consistency

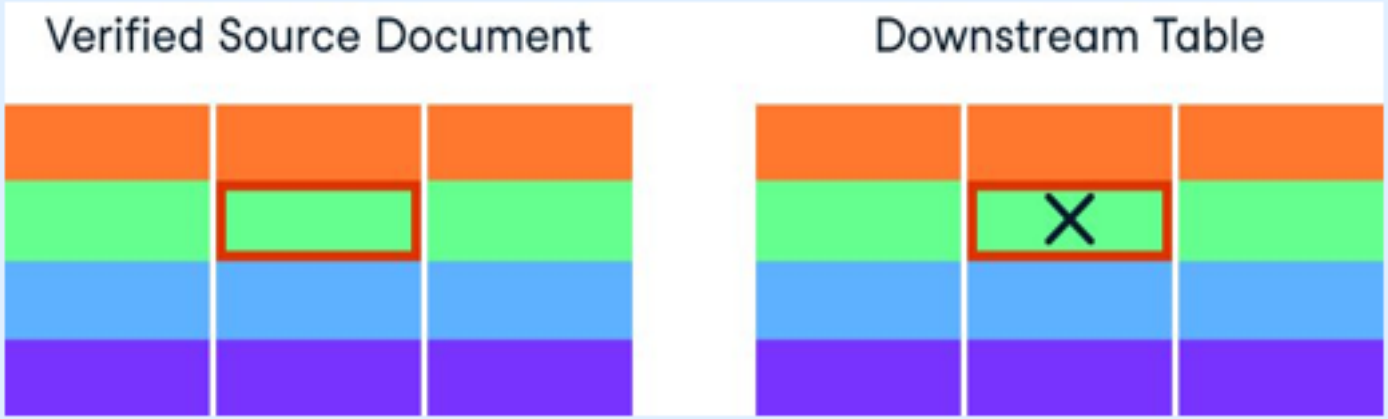
Example

The count of records loaded today must be within +/- 5% of the count of records loaded yesterday.

Count of records in TargetCustomerTable	Record count difference from previous day
10,000,000	4,909,797 ✗
5,090,203	75 ✓
5,090,128	1 ✓

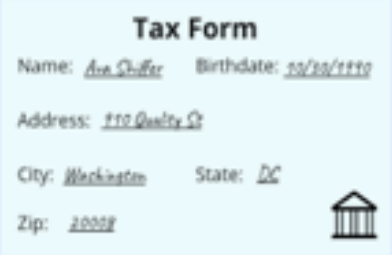
- **Accuracy**

All records in the Customer Table must have accurate Customer Name, Customer Birthdate, and Customer Address fields when compared to the Tax Form.



Accuracy

All records in the Customer Table must have accurate Customer Name, Customer Birthdate, and Customer Address fields when compared to the Tax Form.



CustomerName	CustomerBirthDate	CustomerAddress	CustomerCity	CustomerState	CustomerZip
Ava Shiffer	10/31/1990	910 Quality St	Washington	WA	20008

Example

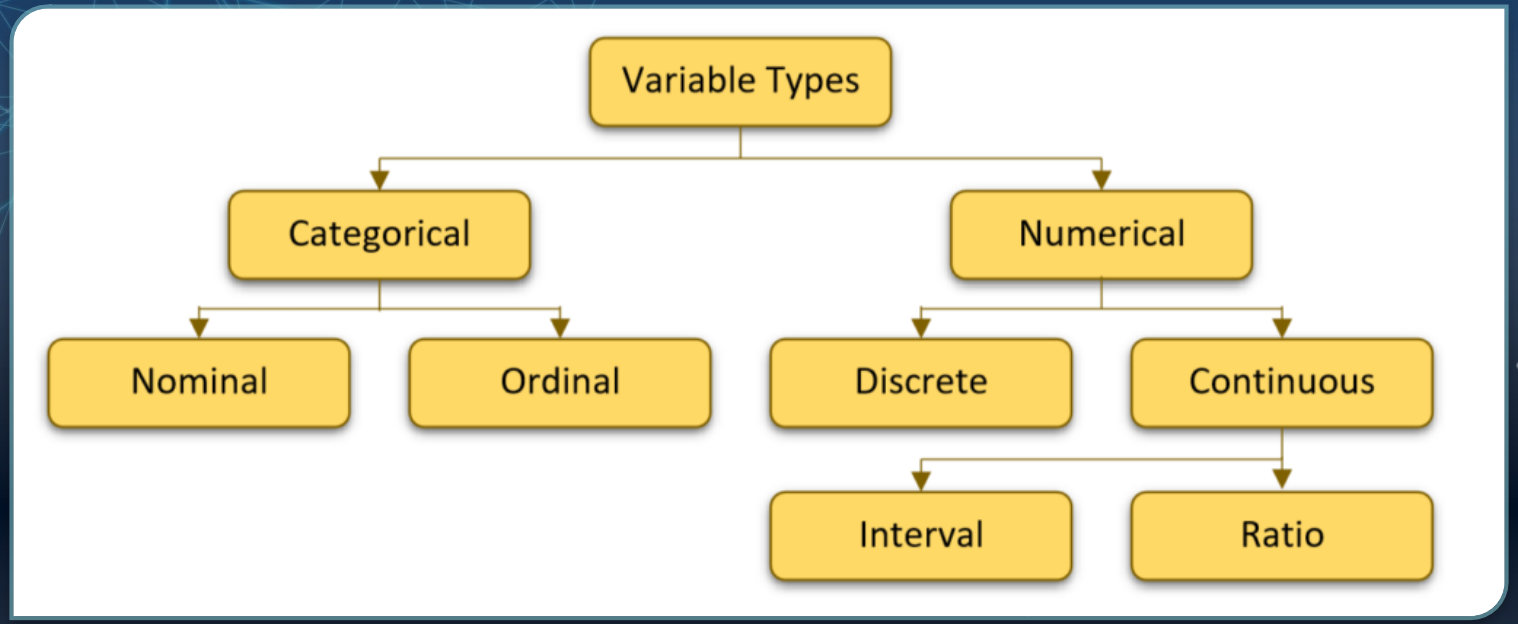
Welcome to the world of
Data Cleaning & data preprocessing

- The dirty data includes things like **incomplete**, **inaccurate**, **irrelevant**, **corrupt** or **incorrectly formatted data**. The process also involves **deduplicating**, or '**deduping**'. This effectively means merging or removing identical data points.

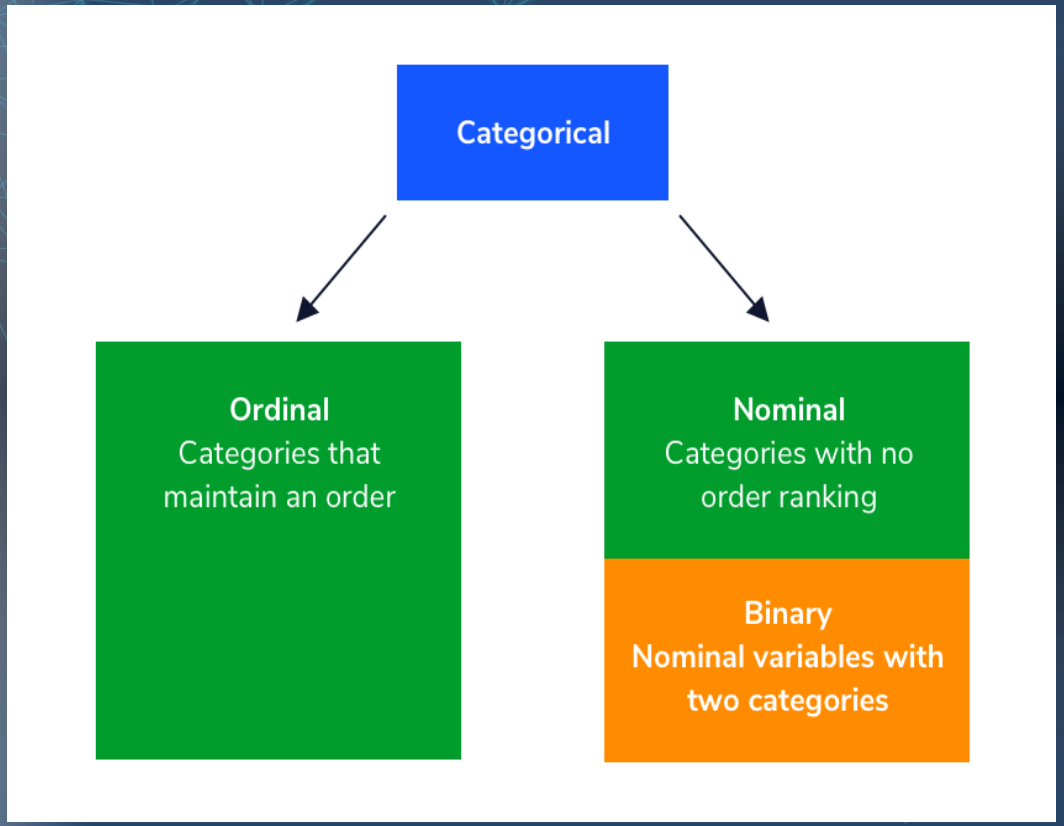
- Note: Data cleaning is time-consuming with great importance comes great time investment. Data analysts spend anywhere from 60-80% of their time cleaning data.

WARNING !

The following concepts and operations will be applied are *NOT SEQUENTIAL*



Types of Attributes



Categorical: Nominal VS. Ordinal

Categorical

Nominal


Pen


Pencil


Eraser


Cow


Dog


Cat

Ordinal


Excellent


Good


Bad


Fantastic


Okay


Don’t Like

ORDINAL DATA

Ordinal data classifies variables into categories which have a natural order or rank.

Examples

School grades

Education level

Seniority level

How is ordinal data analyzed?

- **Descriptive statistics:** Frequency distribution, mode, median, and range
- **Non-parametric** statistical tests

Ordinal Attribute

Test your Understanding

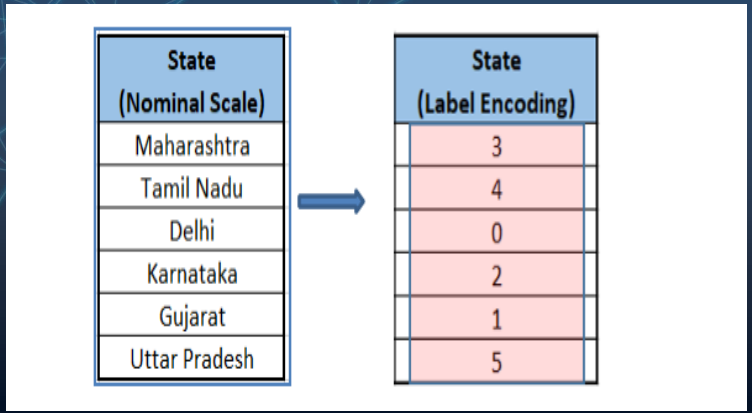
Attribute	Values
Cancer detected	Yes, No
result	Pass , Fail

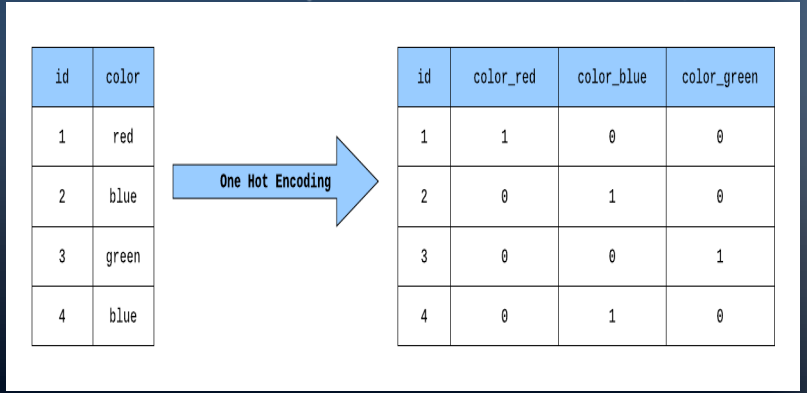
Attribute	Value
Grade	A,B,C,D,E,F
Basic pay scale	16,17,18

Attribute	Value
Profession	Teacher, Business man, Peon
ZIP Code	301701, 110040

Attribute	Values
Colours	Black, Brown, White
Categorical Data	Lecturer, Professor, Assistant Professor

Does Computer Understand text ?





First:
Label Encoding & One-hot Encoding

```
Original Categorical Data: ["Red", "Green", "Blue", "Green", "Red"]


Label Encoding:

---------------

Encoded Data: [0, 1, 2, 1, 0]

Explanation: Each unique category is assigned a unique integer label.
```

Label Encoding

```
One-Hot Encoding:
----------------
Encoded Data:
[1, 0, 0]   # Red
[0, 1, 0]   # Green
[0, 0, 1]   # Blue
[0, 1, 0]   # Green
[1, 0, 0]   # Red
Explanation: A binary vector is used for each category.
```

ONE-HOT Encoding

Label Encoding vs. One-Hot Encoding

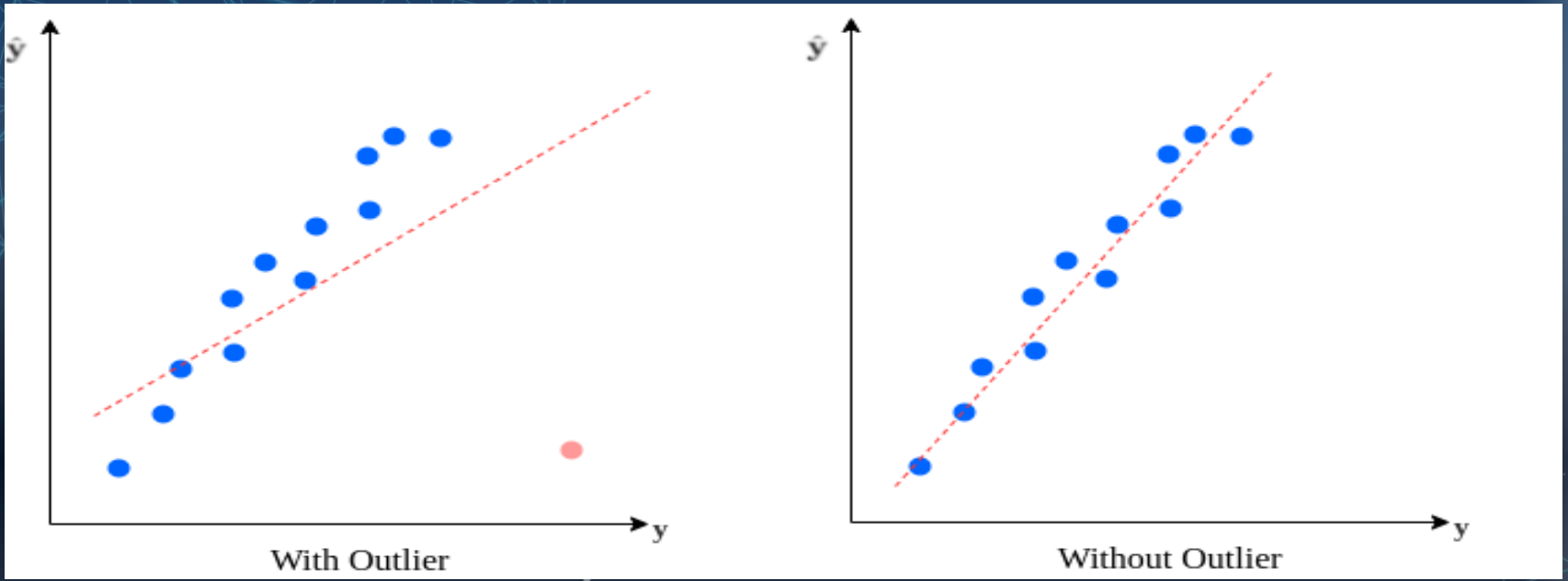
- **Label Encoding** assigns a unique integer label to each category. It results in a single column of integers, which may imply ordinal relationships between the labels that may not exist.

- **One-Hot Encoding** represents each category as a binary vector. It creates multiple columns, one for each category, with binary values (0 or 1) indicating the presence or absence of a category.

What is outliers ?

- **Outliers** are data points that significantly deviate from the rest of the data in a dataset.
- They can be unusually high or low values compared to the majority of the data points.
- Outliers can arise due to various reasons, including measurement errors, data entry mistakes, or genuine extreme values in the underlying phenomenon being studied.

Outliers can have a significant impact on statistical analyses and machine learning models, potentially leading to biased or inaccurate results



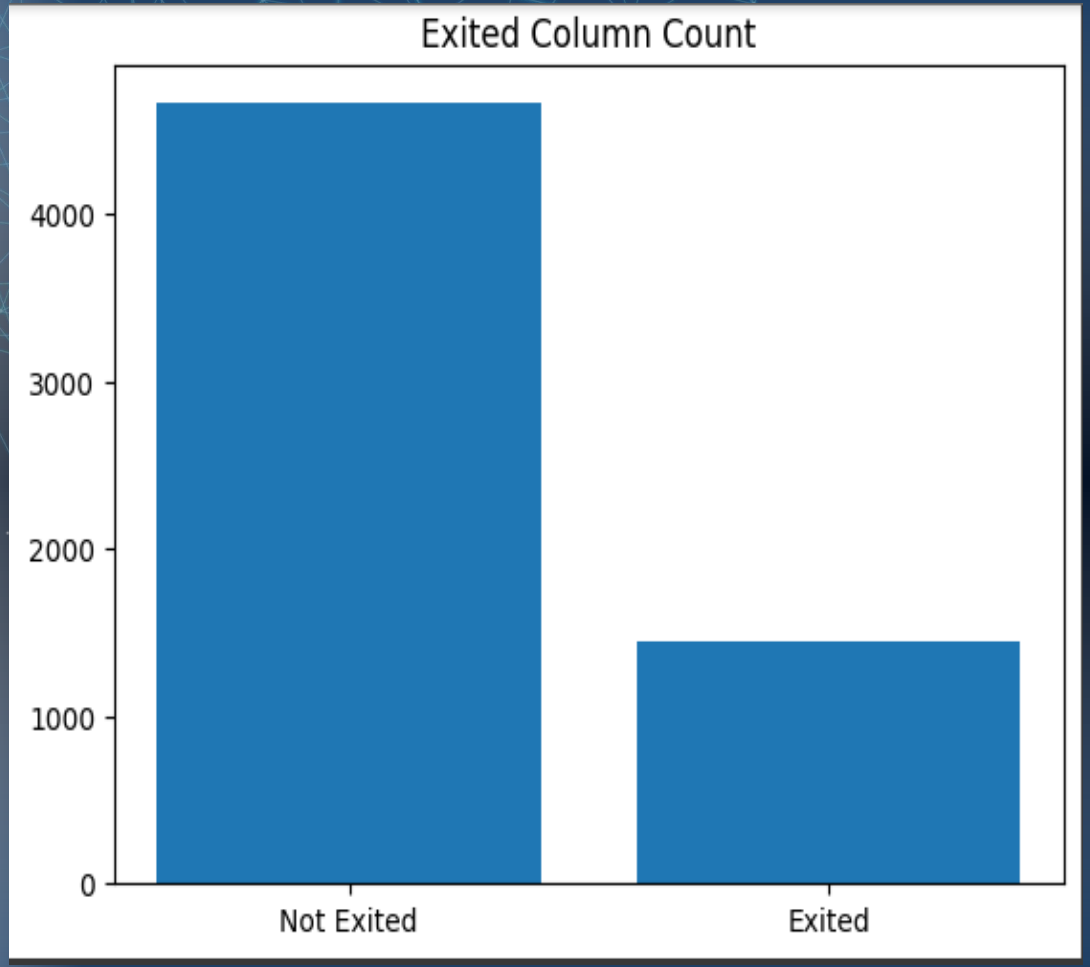
Outliers (Scatter Plot)

1. **Visual Inspection**: Plotting data using histograms, box plots, scatter plots, or other graphical representations can help identify potential outliers.
2. **Z-Score**: The z-score measures how many standard deviations a data point is away from the mean. Data points with z-scores beyond a certain threshold (typically around ±2 to ±3) are considered outliers.
3. **Modified Z-Score**: Similar to the z-score, the modified z-score accounts for the median and uses the median absolute deviation (MAD) as a measure of variability.
4. **Interquartile Range (IQR)**: The IQR is the range between the first quartile (25th percentile) and the third quartile (75th percentile) of the data. Data points outside a certain range beyond the quartiles are considered outliers.

How to detect outliers?

1. **Removal**: Outliers can be removed from the dataset if they are likely to be data entry errors or have a negligible impact on the analysis.
2. **Transformation**: Applying data transformations (e.g., logarithmic, square root) can reduce the impact of outliers and make the data distribution more normal.
3. **Capping or Winsorization**: Capping or replacing extreme values with a predefined threshold can help mitigate the effect of outliers.
4. **Imputation**: Outliers can be replaced with more typical values using statistical imputation methods.
5. **Advanced Models**: Robust statistical models and machine learning algorithms that are less sensitive to outliers can be used.

How to Handle Outliers?



IMBALANCED DATASETS (Bar Charts)

What is Imbalancing ?

- An **imbalanced dataset** refers to a dataset in which the distribution of classes (categories or labels) is significantly skewed or uneven.
- In other words, one class has a much larger number of instances compared to one or more other classes.
- This imbalance can pose challenges for various machine learning algorithms and statistical analyses, particularly those that assume a roughly equal distribution of classes.

1. **Bias in Model Performance**: Algorithms tend to be biased toward the majority class, resulting in poorer performance on the minority class.
2. **Limited Learning**: Algorithms may struggle to learn patterns from the minority class due to the limited number of instances.
3. **Misclassification Costs**: In some applications, misclassifying the minority class may be more costly than misclassifying the majority class.
4. **Evaluation Metrics**: Traditional accuracy may not be an appropriate metric for imbalanced datasets. Other metrics like precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) are more suitable.
5. **Overfitting**: Algorithms may overfit the majority class, resulting in poor generalization to new data.

What is the challenges against Imbalanced datasets ?

1. **Resampling**: This involves either oversampling the minority class, undersampling the majority class, or generating synthetic samples (e.g., using SMOTE - Synthetic Minority Over-sampling Technique).
2. **Cost-sensitive Learning**: Modify the algorithm's cost function to penalize misclassification of the minority class more heavily.
3. **Ensemble Methods**: Techniques like boosting or bagging can help improve the performance of algorithms on imbalanced datasets.

Solutions

4. **Anomaly Detection**: Use specialized anomaly detection algorithms that are designed to handle imbalanced scenarios.
5. **Evaluation Metrics**: Focus on precision, recall, F1-score, and AUC-ROC to evaluate model performance.
6. **Data Augmentation**: Generate additional data instances for the minority class using techniques like text augmentation (in NLP) or image augmentation (in computer vision).
7. **Transfer Learning**: Utilize pre-trained models and fine-tune them for the imbalanced dataset.

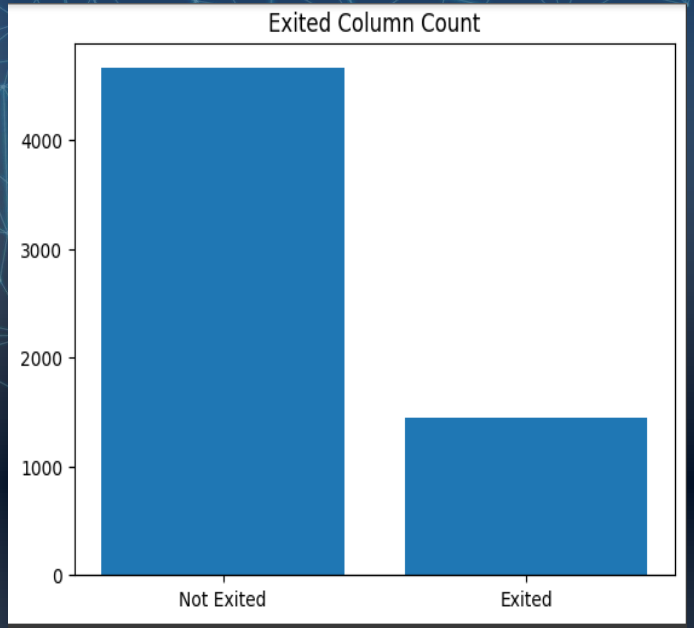
Solutions (Cont.)

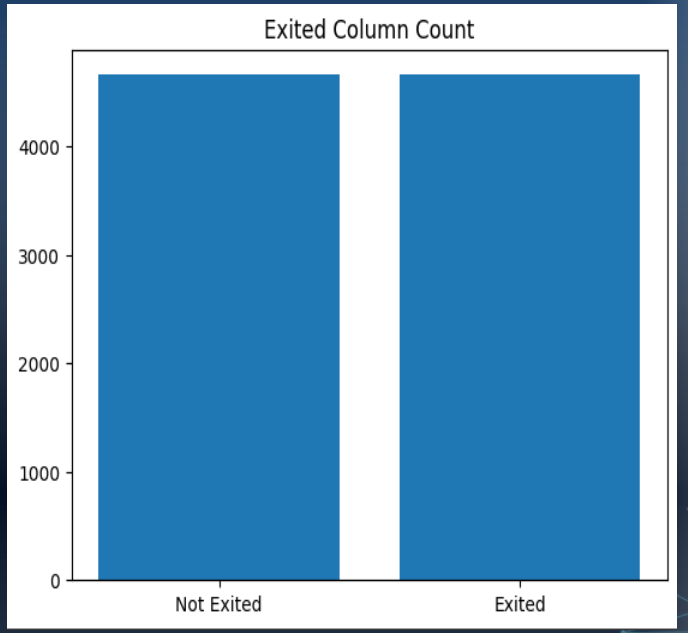
Synthetic-Minority Oversampling Technique (SMOTE Oversampling)

- It is a popular technique used to address the issue of class imbalance in machine learning datasets.
- **SMOTE** is specifically designed to increase the representation of the minority class by generating synthetic samples.
- In a class-imbalanced dataset, the minority class (i.e., the class with fewer instances) is often underrepresented, which can lead to biased model performance and poor generalization.

1. For each instance in the minority class, SMOTE selects k nearest neighbors from the same class. The value of k is a parameter chosen by the user.
2. Synthetic samples are generated by interpolating between the selected instance and its k nearest neighbors. This is done by randomly selecting a neighbor and computing the difference between the feature values of the instance and the neighbor. A random fraction of this difference is added to the instance to create a new synthetic sample.
3. The process is repeated until the desired level of balance is achieved.

SMOTE Algorithm





Before and After balancing

Normalization & Scaling

- **Normalization and scaling are techniques used in data preprocessing to transform the features (variables) of a dataset into a specific range or distribution.**
- **These techniques are important for preparing data before feeding it into machine learning algorithms, as they can improve the performance and convergence of various models.**

Normalization

- **Normalization involves transforming the entire dataset so that each feature has a similar scale.**
- **The goal is to bring all features to a similar range, typically between 0 and 1.**
- **This is particularly useful for algorithms that rely on distances or gradients, such as k-nearest neighbors (KNN) and gradient descent-based optimization algorithms.**

1. **Min-Max Scaling**: Scales the data to a specified range (e.g., [0, 1]) using the formula: `x_normalized = (x - min(x)) / (max(x) - min(x))`.
2. **Z-Score Normalization (Standardization)**: Transforms the data to have a mean of 0 and a standard deviation of 1 using the formula: `x_standardized = (x - mean(x)) / std(x)`.

Types of Normalization

Scaling

- **Scaling, on the other hand, focuses on adjusting the range of features without necessarily changing their distribution or statistical properties.**

- **Scaling can be important for algorithms that use distance measures but don't necessarily require a specific range for input features.**

1. **Min-Max Scaling**: As mentioned above, this technique scales features to a specific range.
2. **Z-Score Scaling (Standardization)**: As mentioned above, this technique standardizes features to have a mean of 0 and a standard deviation of 1.
3. **Robust Scaling**: This technique is similar to Z-score scaling but uses the median and interquartile range, making it more robust to outliers.
4. **Max Absolute Scaling**: Scales features by dividing them by their maximum absolute value, resulting in values within the range [-1, 1].
5. **Unit Vector Scaling (Normalization)**: Scales features to have a Euclidean norm (magnitude) of 1, which can be useful for algorithms that rely on vector distances.

Types of Scaling

- Use **Normalization** when you want to ensure that all features are on a similar scale, which can be important for distance-based algorithms.
- Use **Scaling** when you want to adjust the range of features without necessarily changing their distribution, and the specific scale of features isn't critical.

Normalization VS. Scaling

Pearson Correlation

- **Pearson's correlation coefficient** or Pearson's r, is a statistical measure that quantifies the linear relationship between two continuous variables.

- It assesses how closely the data points of two variables align on a straight line. The coefficient ranges from -1 to 1, where:

  - A value of +1 indicates a perfect positive linear relationship, meaning that as one variable increases, the other also increases proportionally.

  - A value of -1 indicates a perfect negative linear relationship, meaning that as one variable increases, the other decreases proportionally.

  - A value of 0 indicates no linear relationship between the variables; they are independent of each other.

Pearson Correlation (Cont.)

- In essence, Pearson correlation helps to determine if there is a consistent pattern in th two variables change in relation to each other.

- It's important to note that Pearson correlation specifically measures linear relationship the relationship between the variables is not linear, the correlation coefficient ma accurately capture their association.

The formula for calculating Pearson correlation coefficient between two variables, X and Y, based on their sample data, is as follows:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

Where:

- X_i and Y_i are individual data points of variables X and Y, respectively.
- $\bar{X}$ and $\bar{Y}$ are the means (averages) of variables X and Y, respectively.

Pearson Correlation (Cont.)

	0	1	2	3	4	5	6	7	8	9
0	1	0.35	0.4	0.46	0.073	-0.23	-0.73	0.48	-0.44	0.015
1	0.35	1	-0.28	0.57	-0.29	0.38	-0.36	0.64	0.25	0.19
2	0.4	-0.28	1	-0.52	0.15	-0.14	-0.093	0.016	-0.43	-0.38
3	0.46	0.57	-0.52	1	-0.23	-0.23	-0.48	0.47	0.28	0.45
4	0.073	-0.29	0.15	-0.23	1	-0.1	-0.15	-0.52	-0.61	-0.19
5	-0.23	0.38	-0.14	-0.23	-0.1	1	-0.03	0.42	0.21	0.095
6	-0.73	-0.36	-0.093	-0.48	-0.15	-0.03	1	-0.49	0.38	-0.35
7	0.48	0.64	0.016	0.47	-0.52	0.42	-0.49	1	0.38	0.42
8	-0.44	0.25	-0.43	0.28	-0.61	0.21	0.38	0.38	1	0.15
9	0.015	0.19	-0.38	0.45	-0.19	0.095	-0.35	0.42	0.15	1

Questions


Thank You