Pre Processing

Sooroush Riazi

OUTLINE

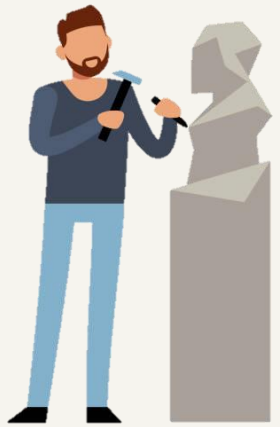
- Introduction to Data Preprocessing
- Data Cleaning and Handling Missing Values
  - Strategies for Handling Missing Values
- Detecting and Handling Outliers
  - Z Score
  - IQR
  - Visualization
- Data Integrity
  - Duplicated Records
  - Feature Inconsistencies:

OUTLINE

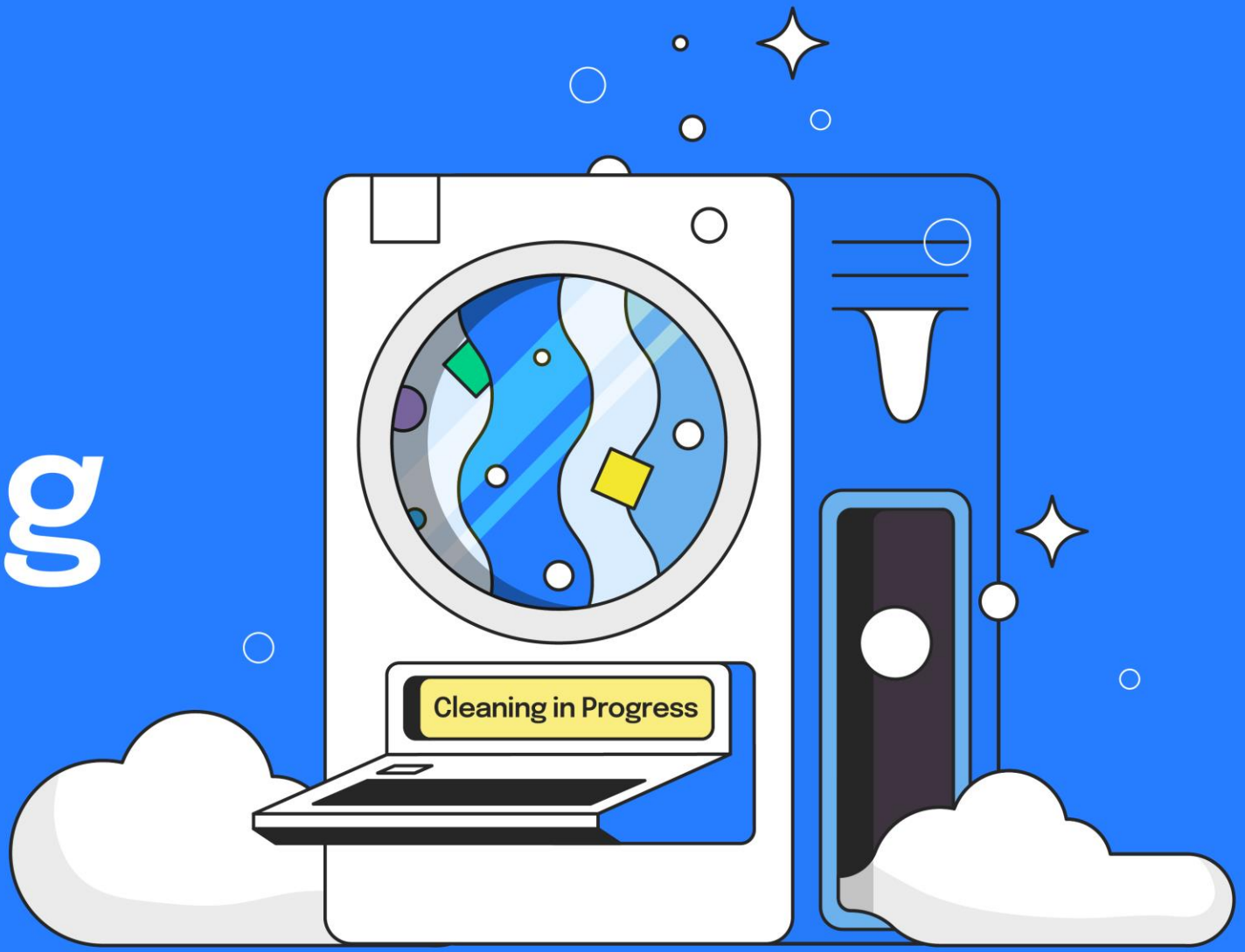
- Data Transformation Techniques
  - Why Scaling and Normalization
  - Scaling and Normalization
  - Handling Categorical Data (Covered in Feature engineering)
- Feature Engineering
  - Feature Creation
  - Feature Selection
  - Feature Extraction
  - Encoding Categorical Features
- Resampling
  - OverSampeling
  - Undersampeling

Introduction to Data Preprocessing

- Welcome to the Data Preprocessing course! In the exciting field of data science, working with raw data is like having a block of marble – full of potential, but in need of sculpting to reveal its true beauty. This is where data preprocessing comes into play.



Data Cleaning



Data Cleaning and Handling Missing Values

- Just as a painter prepares their canvas before creating a masterpiece, data scientists must meticulously clean and repair their datasets to ensure accurate and meaningful analyses.

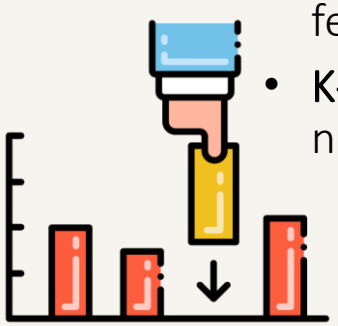
- Data, like any other raw material, can come with imperfections. These imperfections might manifest as **missing values, outliers, or inconsistencies.**

Handling Missing Values

- Missing data is a common challenge in real-world datasets. It can arise due to various reasons.
  - errors during data collection
  - system failures
  - participants' reluctance to provide certain information
- Regardless of the cause, missing data can significantly impact the validity of our analyses.

Strategies for Handling Missing Values

- Deletion
  - **Listwise Deletion:** Removing entire rows with missing values. While simple, it might lead to loss of valuable information.
  - **Pairwise Deletion:** Removing specific columns with missing values while retaining other columns. This can introduce bias if not handled carefully.
- Imputation
  - **Mean, Median, Mode:** Replacing missing values with the mean, median, or mode of the corresponding feature. Simple but effective, these methods maintain the dataset's general characteristics.
  - **K-Nearest Neighbors (KNN) Imputation:** Predicting missing values based on the values of k-nearest neighbors in a multidimensional space.



- i dont want these features: Drop

```
data = data.drop(['Number','Wind_Chill(F)','ID','Description'], axis=1)
```

- Filling Missing Value and Droping UseLess Features

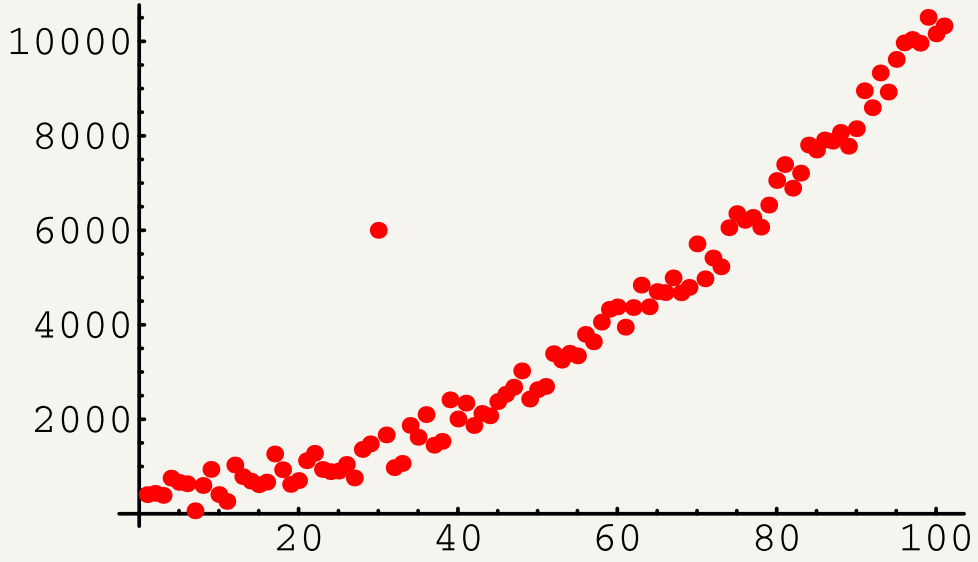
```
data['City'] = data['City'].fillna(method='ffill')
data['Weather_Condition'] = data['Weather_Condition'].fillna(method='ffill')
data['Wind_Direction'] = data['Wind_Direction'].fillna(method='ffill')
data['Wind_Speed(mph)'] = data['Wind_Speed(mph)'].fillna(method='ffill')
data['Visibility(mi)'] = data['Visibility(mi)'].fillna(method='ffill')
data['Pressure(in)']=data['Pressure(in)'].fillna(method='ffill')
data['Humidity(%)']=data['Humidity(%)'].fillna(method='ffill')
```

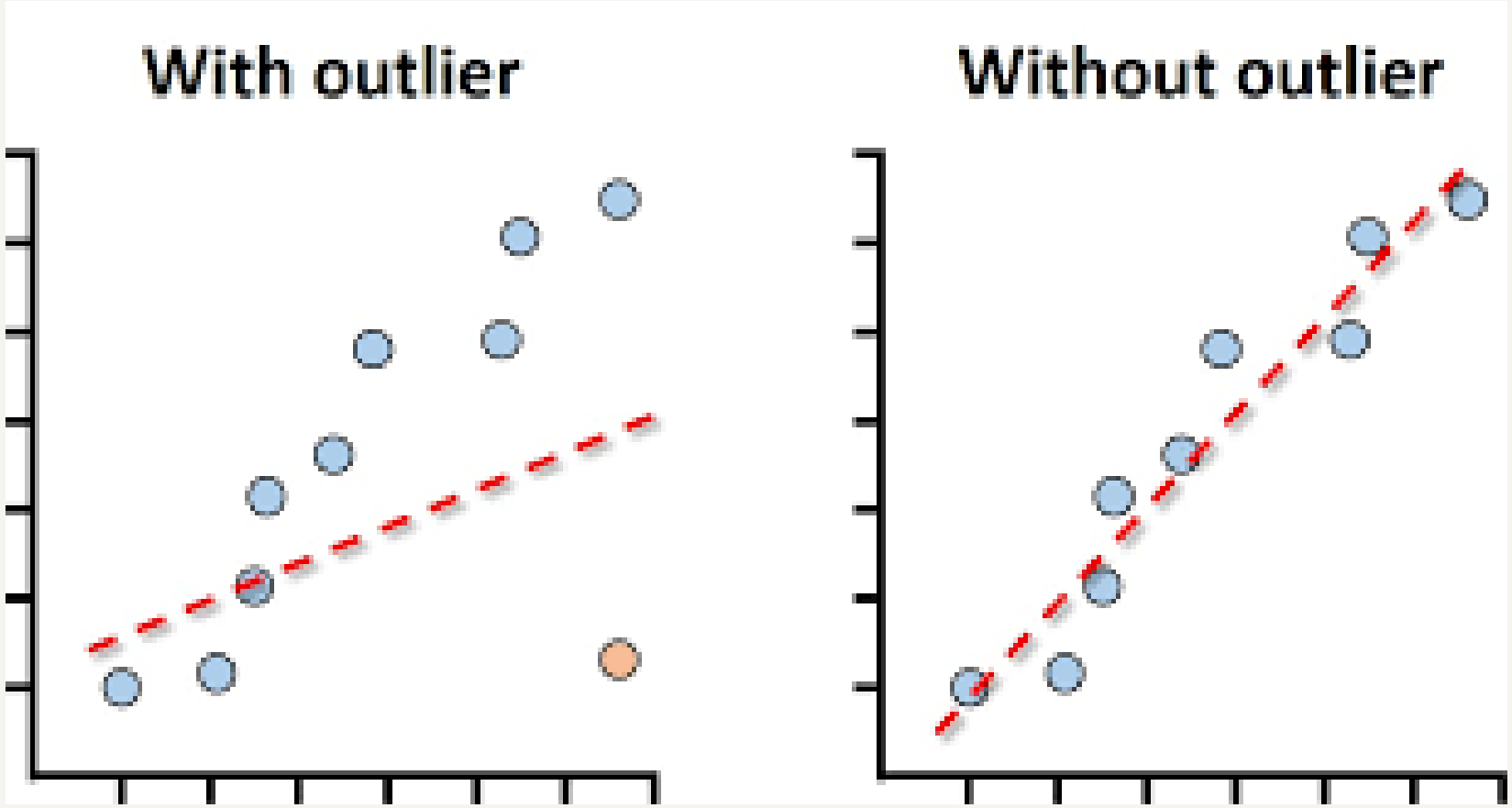
Think !

Think about power of ai and use your creation to suggest a new way for handling missing values.

Detecting and Handling Outliers

- Outliers are data points that deviate significantly from the rest of the dataset. They can distort analysis results and adversely affect model performance. Identifying and handling outliers is crucial to ensure the accuracy and reliability of our analyses.




With outlier
Without outlier

Outlier Detection and handling

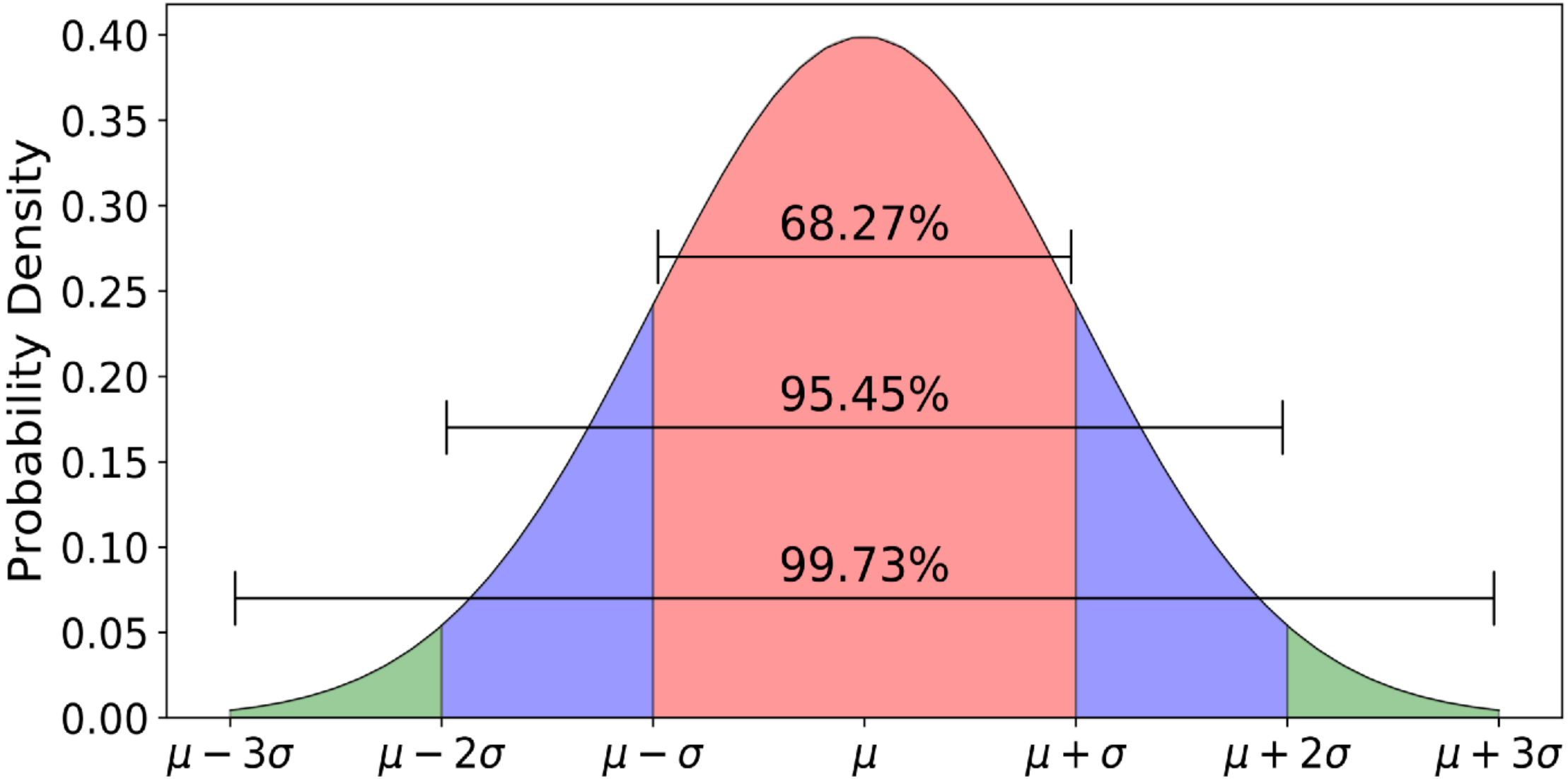
- **Z-Score:** Calculating the z-score for each data point to identify those that fall beyond a certain threshold.
- **Interquartile Range (IQR):** Using the IQR to identify and remove data points beyond a specified range.
- **Visualization Techniques:** Creating box plots, scatter plots, or histograms to visually identify outliers.

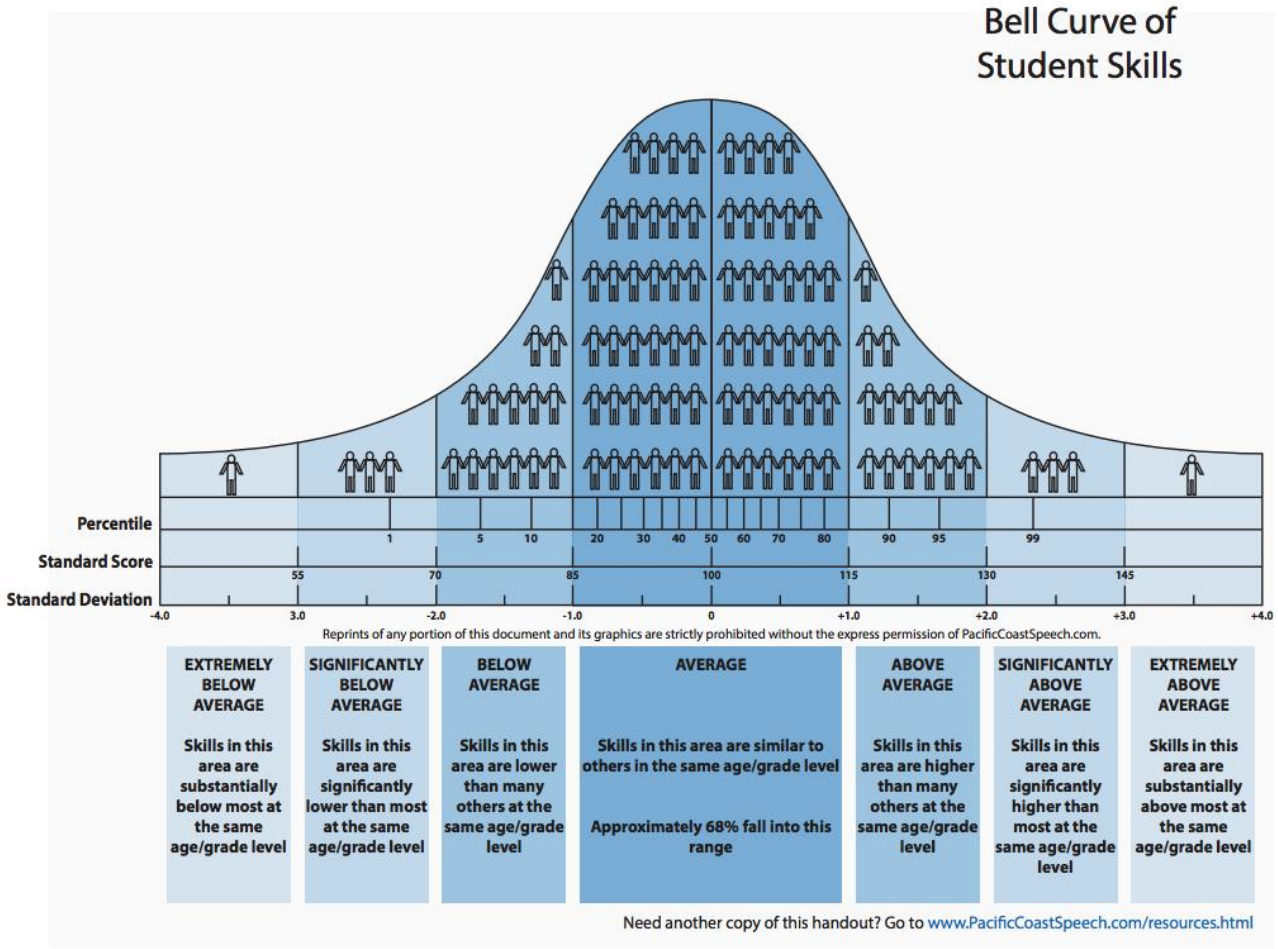
Z-Score

A z-score less than -3 or greater than 3 is often used as a threshold for identifying outliers. These z-scores correspond to data points that are more than 3 standard deviations away from the mean in a normal distribution.

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

- x: The individual data point you want to calculate the z-score for.
- μ: The mean (average) of the dataset.
- σ: The standard deviation of the dataset.

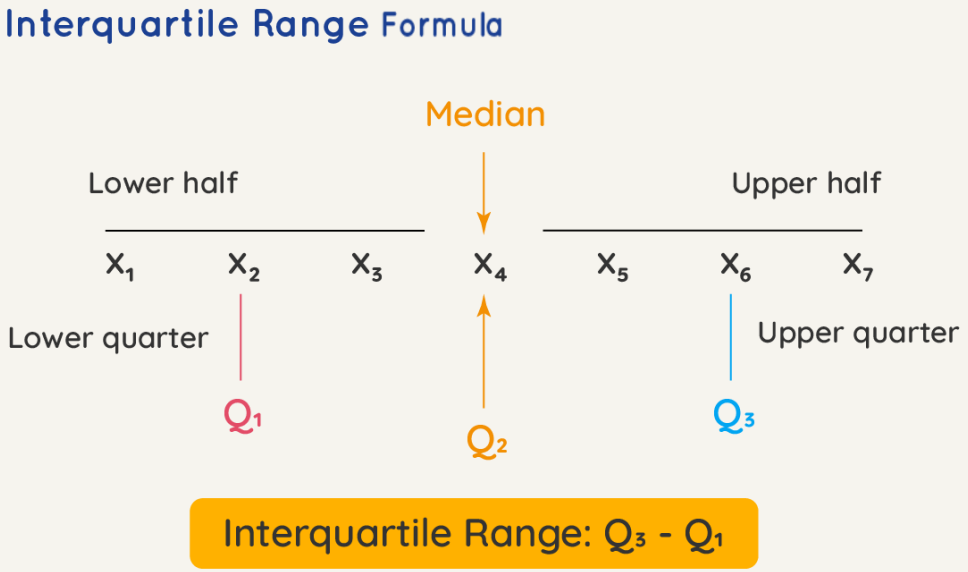

Probability Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
68.27%
95.45%
99.73%
$\mu - 3\sigma$
$\mu - 2\sigma$
$\mu - \sigma$
μ
$\mu + \sigma$
$\mu + 2\sigma$
$\mu + 3\sigma$



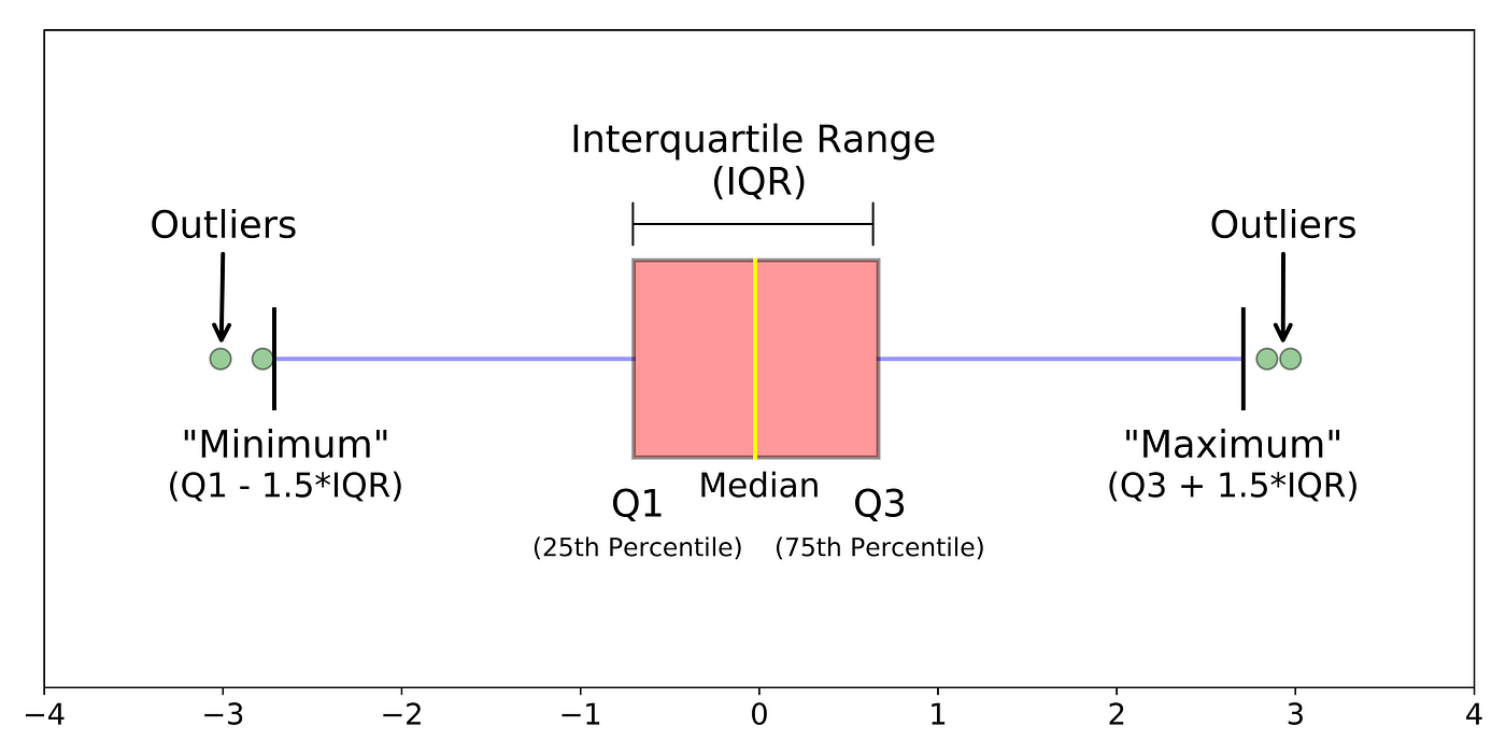
- Keep in mind that the z-score assumes a normal distribution of data. If your data is not normally distributed, other techniques, such as the interquartile range (IQR), may be more appropriate for identifying outliers.

Interquartile Range (IQR):

- The Interquartile Range (IQR) is a statistical measure that represents the spread or variability of a dataset. It is particularly useful for identifying and handling outliers in non-normally distributed data. which divide the data into four equal parts.

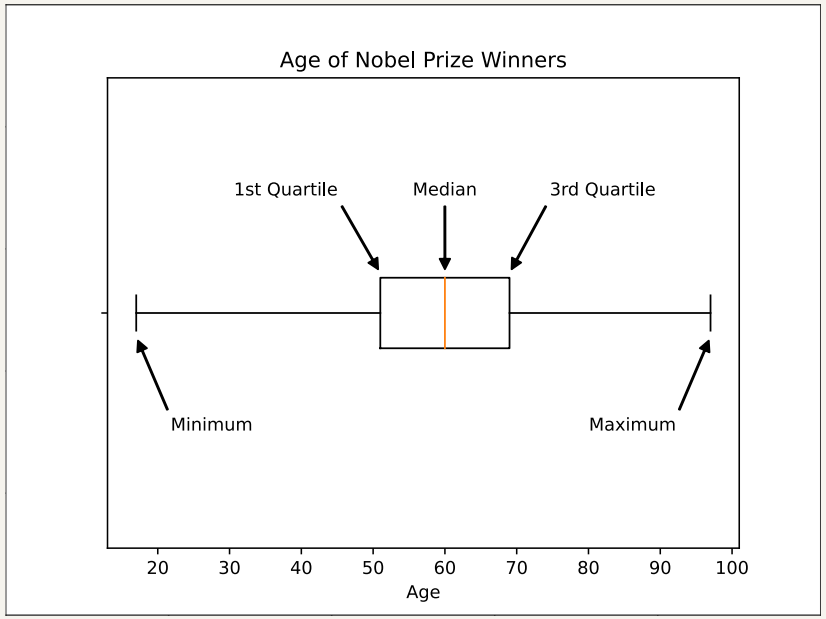


IQR for Outlier



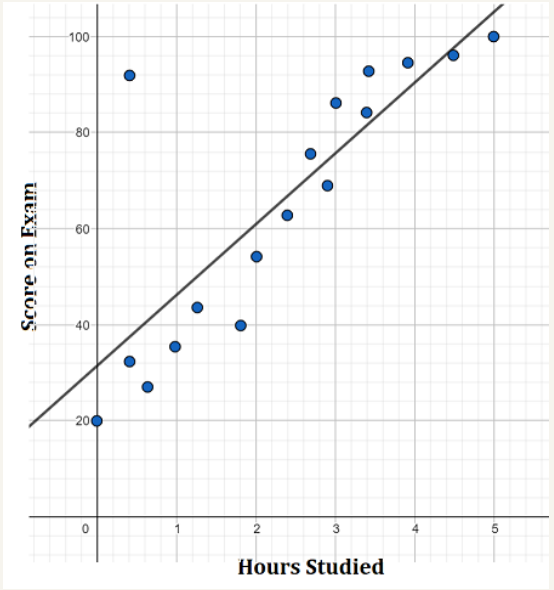
Visualization Techniques

- Box Plots: Outliers are often shown as individual points beyond the "whiskers" (lines extending from the box), which are typically set at a certain multiple of the interquartile range (IQR).



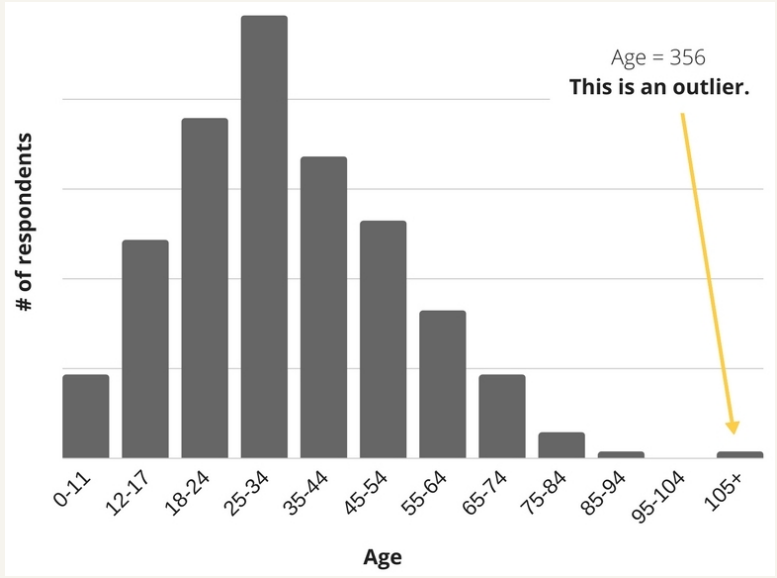
Visualization Techniques

- **Scatter Plots:** Scatter plots are effective for identifying outliers in **bivariate data**. Outliers may appear as points that deviate significantly from the general pattern of the data.



Visualization Techniques

- **Histograms:** Histograms display the frequency distribution of data. Unusual spikes or gaps in the histogram can indicate the presence of outliers.



Advanced Technics

PCA (Principal Component Analysis) for Outlier Detection:

- High-dimensional data
- Visualizing outliers

Autoencoders

NMF (Non-Negative Matrix Factorization) for Outlier Detection:

- robust to noise and minor variations.

Data Integrity

- Duplicate records:
  - can also compromise the integrity of our analyses. Identifying and removing duplicate entries is an essential step to ensure that our dataset accurately represents the underlying reality.

- Feature Inconsistencies:
  - Feature inconsistencies occur when the values or attributes within a specific feature deviate from the expected patterns, leading to discrepancies or conflicts that can affect data quality and analysis.

Feature Inconsistencies:

- Causes:
  - Errors during data collection, entry, or integration from multiple sources.
  - Mismatched units, scales, or formats in numerical features.
  - Contradictory or conflicting information within categorical features.
- **Impact:** Feature inconsistencies can mislead analyses, introduce noise, and hinder accurate model predictions.

Examples of Feature Inconsistencies:

- 1. Date Format Inconsistency:
  - Suppose you're analyzing a dataset that includes a "Date of Birth" feature. Inconsistent date formats (e.g., "MM/DD/YYYY" vs. "YYYY-MM-DD") can lead to data entry errors and confusion during analysis.
- Example:
  - "Date of Birth": "05/15/1990" (MM/DD/YYYY)
  - "Date of Birth": "1990-05-15" (YYYY-MM-DD)
- 2. Unit Mismatch:
  - Imagine a dataset with a "Product Weight" feature recorded in pounds and another feature, "Product Price," recorded in euros. Such inconsistencies in units can lead to incorrect analysis and model predictions.
- Example:
  - "Product Weight": 10 pounds
  - "Product Price": 20 euros

Examples of Feature Inconsistencies:

3. Categorical Contradiction:

- Consider a dataset with a "Customer Gender" feature and another "Is Pregnant" feature. If the "Customer Gender" is recorded as "Male" but "Is Pregnant" is marked as "True," this is a categorical contradiction that needs resolution.

4. Inconsistent Spelling:

- In textual data, inconsistent spelling or variations of the same word can create discrepancies. For instance, "USA," "U.S.A.," and "United States" may refer to the same entity.

5. Missing Data Handling Inconsistencies:

- Different missing data indicators (e.g., "NA," "Missing," "None") within the same feature can complicate data preprocessing and model training.

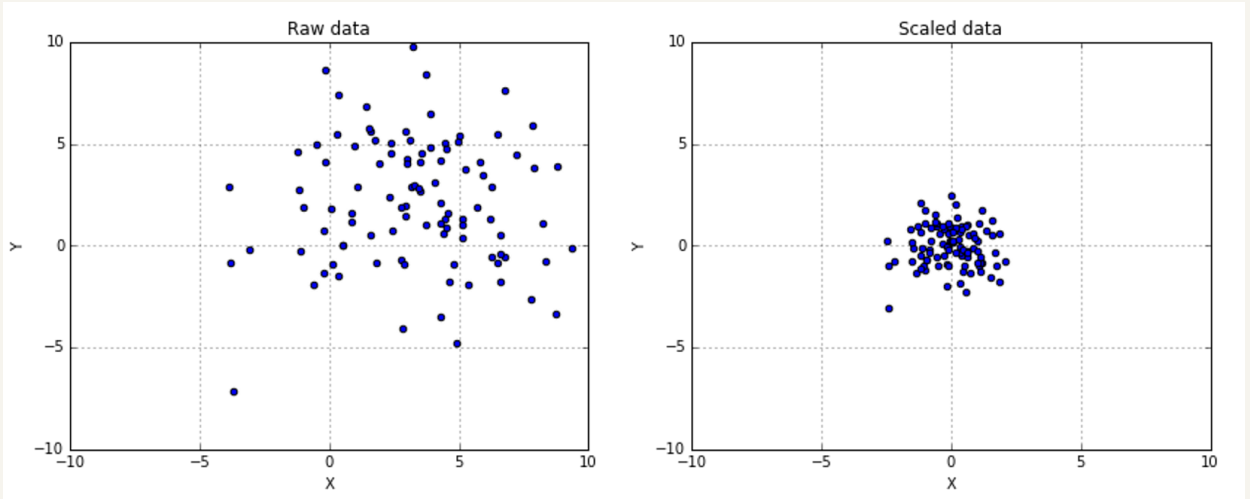
- Example:
  - "Income": NA
  - "Income": Missing

Data Transformation Techniques

- Scaling and Normalization
- Handling Categorical Data (Covered in Feature engineering)

Why Scaling and Normalization

- Scaling is a fundamental data transformation technique used to bring numerical features to a common scale. It's an essential step in data preprocessing, especially when working with algorithms that are sensitive to the magnitude of features. Scaling ensures **that each feature contributes equally to the analysis and prevents features with larger values from dominating the learning process.**



Why Scaling and Normalization

1. **Convergence Speed:** Algorithms like gradient descent converge faster when features are on a similar scale, as they take more balanced steps toward the optimal solution.
2. **Distance-Based Algorithms:** Algorithms like k-means clustering and hierarchical clustering are affected by the distance between data points. Scaling ensures that features with larger magnitudes do not dominate the distance computation.
3. **Regularization:** Regularization techniques, such as Ridge and Lasso regression, penalize large coefficients. Scaling prevents certain features from being overly penalized due to their larger values.
4. **Model Performance:** Scaling can improve the performance and stability of algorithms like support vector machines (SVMs) and neural networks.

Scaling Methods

- Min-Max Scaling (Normalization):
  - Scales features to a specified range (usually [0, 1]).
  - Suitable when features have a defined minimum and maximum, and you want to maintain interpretability.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Standard Normalization:
  - also known as Z-score normalization or standardization, is a data transformation technique used to scale numerical features so that they have a mean of 0 and a standard deviation of 1. It's a common method in data preprocessing and is particularly useful when dealing with features that have different units of measurement or scales.

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}}$$

Feature Engineering

Feature engineering is a critical phase in the data preprocessing journey that involves crafting and refining features to optimize the performance of machine learning models. In this session, we will explore the art and science of feature engineering, delving into techniques that empower us to extract meaningful information from raw data, improve model interpretability, and enhance overall predictive accuracy.

- Feature Creation
- Feature Selection
- Feature Extraction
- Encoding Categorical Features

Feature Creation

- **Ratio Features:** Creating new features by calculating ratios between numerical variables. For example, calculating a debt-to-income ratio.
- **Aggregation:** Aggregating data to create summary statistics, such as mean, median, and sum, for specific groups or time intervals.
- **Date/Time Features:** Extracting components like day, month, year, and time of day from timestamps.

Feature Selection

- Challenges of high-dimensional data.
- Enhance model performance and interpretability.

1. **Recursive Feature Elimination (RFE)**: Iteratively removing least important features.
2. **Correlation Analysis**: Analyzing feature correlations.
3. **Feature Importance from Tree-Based Models**: Assessing feature importance using ensemble models.

Encoding Categorical Features

- Transform categorical data into numerical form for models.

1. One-Hot Encoding:
   - Convert categories to binary vectors.
   - Each category becomes a separate column
2. Label Encoding:
   - Assign unique integer labels to categories.
   - Suitable for ordinal categorical variables.

Label Encoding VS One Hot Encoding

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

One-Hot Encoding

- **Advantages:**
- Prevents ordinality issues: One-hot encoding avoids falsely introducing ordinal relationships between categories.
- Compatible with most algorithms: Many machine learning algorithms expect numerical input, making one-hot encoding a suitable choice.
- **Limitations:**
- Dimensionality increase: One-hot encoding increases the dimensionality of the dataset, especially for variables with many categories.
- Collinearity: The new columns might be correlated, which can affect the performance of certain algorithms.

Label Encoding

- Advantages:
  - Compact representation: Label encoding reduces the dimensionality of the categorical variable to a single column of integers.
  - Preserves ordinality: Suitable for ordinal categorical variables where order matters.
- Limitations:
  - Misleading magnitude: Some algorithms may misinterpret the integer labels as ordinal magnitudes, leading to incorrect assumptions.
- When to Use:
  - For ordinal categorical variables where the order of categories has significance.

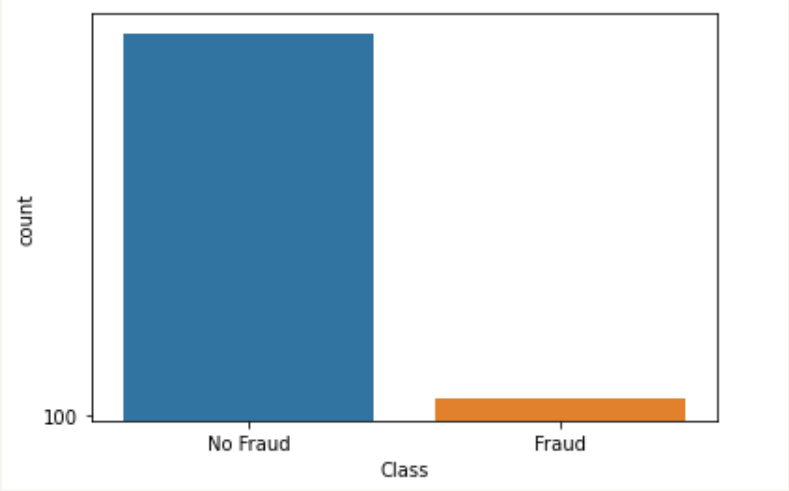
Feature Extraction

- Transforming high-dimensional data into lower-dimensional representation.
- Reduce dimensionality, remove noise, capture patterns.

1. Principal Component Analysis (PCA):
    1. Identify orthogonal components with maximum variance.
    2. Eigenvalues, eigenvectors, explained variance.
2. Non-Negative Matrix Factorization (NMF):
    1. Decompose non-negative data into additive components.
    2. Application in topic modeling, image processing.

Imbalanced Data

- **Definition:** Class imbalance occurs when the distribution of classes in a dataset is uneven, with one class significantly outnumbering the others.
- **Impact**: Class imbalance can lead to biased model predictions, poor generalization, and inaccurate results, as models tend to favor the majority class.



Resampling

- Resampling is a critical technique in data preprocessing used to address class imbalance and enhance the performance of machine learning models. This session explores the intricacies of resampling methods and how they can mitigate issues stemming from imbalanced datasets.
- Oversampling
- Undersampling

Oversampling

- **Oversampling**: Oversampling involves increasing the representation of the minority class by generating synthetic samples or replicating existing ones.
  - Potential overfitting on duplicated samples.

- **Methods:**
  1. Random Oversampling: Replicate minority class samples randomly.
  2. Synthetic Minority Over-sampling Technique (SMOTE): Generate synthetic samples along line segments between existing minority class samples.

Undersampling

- **Oversampling**: Undersampling involves reducing the representation of the majority class by removing samples.
  - Risk of underfitting due to reduced training data.

- **Methods:**
  1. Random Undersampling: Randomly remove majority class samples.
  2. Edited Nearest Neighbors (ENN): Remove majority class samples that are misclassified by their k-nearest neighbors.

Evaluation Considerations !

Real-world Challenges and Best Practices

Data

```
Country_data = pd.read_csv("Country-data.csv")
Country_data.head()
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6					
2	Algeria	27.3	38.4	4.17	31.4					
3	Angola	119.0	62.3	2.85	42.9					
4	Antigua and Barbuda	10.3	45.5	6.03	58.9					

Data US_Accidents:

```
data = pd.read_csv('US_Accidents.csv')
```
✓ 19.9s

```
data
```
✓ 1.9s

ID	Severity	Start_Time	End_Time	Start_Lng	End_Lat	End_Lng	Distance(mi)	Description	...	Roundabout
			2016-02-			031870	3.230	Between Sawmill Rd/Exit 20 and OH-315/Olentang...	...	False
						048730	0.747	At OH-4/OH-235/Exit 41 - Accident.	...	False
						523960	0.055	At I-71/US-50/Exit 1 - Accident.	...	False

Persian Car Data

```
data = pd.read_csv('stuff.csv')
```
✓ 0.0s

```
data
```
✓ 0.0s

	car	province	city	price	km_count	color	status	inside_color	chassis_type	gearbox	cylinder_count	fuel_type	phone
0	اتوماتیک S5 1396 خرید جک	تهران	تهران	تومان۱,۰۳۰,۰۰۰,۰۰۰	۱۰۰,۰۰۰	سفید	بدون رنگ	مشکی	شاسی بلند	اتوماتیک	سیلندر 4	بنزین	۰۹۱۲ ۷۳۱
1	خرید ام وی ام 530 1391	تهران	تهران	تومان۲۲۰,۰۰۰,۰۰۰	۲۲۰,۰۰۰	نقره ای	بدون رنگ	کرم	صندوق دار	دنده ای	سیلندر 4	بنزین	۰۹۱۲ ۲۹
2	دنده ای H230 1397 خرید برلیانس	مازندران	نوشهر	تومان۲۶۵,۰۰۰,۰۰۰	۱۰۰,۰۰۰	سفید	بدون رنگ	مشکی	صندوق دار	دنده ای	سیلندر 4	بنزین	۰۹۳۸ ۵۹
3	اتوماتیک H320 1.5 1395 خرید برلیانس	البرز	کرج	تومان۵۷۰,۰۰۰,۰۰۰	۱۳۷,۰۰۰	نوک مدادی	یک لکه رنگ	کرم	هاچ بک	اتوماتیک	سیلندر 4	بنزین	۰۹۱۲ ۵۶
4	خرید دنا ساده 1396	تهران	تهران	تومان۵۰۰,۰۰۰,۰۰۰	۴۰,۰۰۰	مشکی	بدون رنگ	مشکی	صندوق دار	دنده ای	سیلندر 4	بنزین	۰۹۱۲ ۷۰۲
...	...	...	...	...	...	...	...	...	...	...	...	...	
1327	خرید پراید صندوق دار ساده بنزینی 1389	تهران	تهران	تومان۱۸۹,۰۰۰,۰۰۰	۱۹۵,۰۰۰	طلائی	بدون رنگ	طوسی	صندوق دار	دنده ای	سیلندر 4	بنزینی	۰۹۰۱ ۴۰
1328	خرید هوندا CR-V 2016	تهران	تهران	توافقی	۲۸,۰۰۰	آلبالویی	بدون رنگ	شتری	شاسی بلند	اتوماتیک	سیلندر 4	بنزین	۰۹۱۲ ۱۰

Innovation Center for Data Science

eds

Special Thanks

For gathering a relevant data and Designing the Summer School

Innovation Center for Data Science

eds3

THANK YOU FOR YOUR ATTENTION

WISH YOU LUCK

Innovation Center for Data Science | eds³ | **Your Journey is beginning**