ASSIGNMENT - 4

1. Data Quality Issues in Analytics Projects

Common data quality issues include:

- **Missing Data**: Data entries that are null or empty, possibly due to data collection or transmission errors.
- **Duplicate Data**: Repeated records that can skew analysis.
- **Inconsistent Data**: Mismatched formats (e.g., “NY” vs. “New York”), varying units, or naming conventions.
- **Inaccurate Data**: Values that are wrong or implausible due to faulty sensors, human error, or outdated information.
- **Outliers**: Extreme values that may distort statistical analysis.
- **Data Entry Errors**: Typos, mislabeling, or invalid formats.
- **Imbalanced Data**: In classification problems, one class is significantly overrepresented.
- **Irrelevant Features**: Data columns that do not contribute meaningfully to the problem being solved.

---

2. Handling Missing Data

Methods & When to Use:

1. **Deletion Methods**:
    - **Listwise Deletion**: Remove rows with missing data.
        - *When*: Data is missing completely at random (MCAR) and dataset is large.
    - **Column Deletion**: Remove columns with too many missing values.
        - *When*: Feature has >50% missing data and low importance.
2. **Imputation Methods**:

- **Mean/Median/Mode Imputation**:
  - *When*: Data is missing at random; use mean for symmetric data, median for skewed data.
- **Forward/Backward Fill (for Time Series)**:
  - *When*: Time series data; continuity is more important.
- **K-Nearest Neighbors (KNN) Imputation**:
  - *When*: Data is not missing completely at random; similar instances can predict missing values.
- **Multivariate Imputation (e.g., MICE)**:
  - *When*: Complex missing data with interdependent features.
- **Model-Based Imputation** (Regression, XGBoost, etc.):
  - *When*: Need accurate predictions for missing values in key features.

3. **Use Flags**:
   - Add a binary column indicating missingness.
     - *When*: Missingness itself may carry information.

---

3. Label Encoding vs. One-Hot Encoding vs. Ordinal Encoding

Encoding Type	Description	When to Use	Example
Label Encoding	Assigns a unique integer to each category.	For tree-based models (e.g., Random Forests); when no ordinal relationship.	`Red=0, Green=1, Blue=2`
One-Hot Encoding	Converts each category into binary columns.	For non-ordinal data; used with linear models or neural networks.	`Color_Red=1, Color_Green=0, Color_Blue=0`
Ordinal Encoding	Maps categories to integers based on order.	When categories have meaningful order (e.g., levels).	`Low=1, Medium=2, High=3`

⚠️ Use One-Hot for nominal (no order), Ordinal Encoding for ordered categories, and Label Encoding sparingly (only with models that can handle categorical integers appropriately).

---

4. Importance of Data Scaling: Normalization vs. Standardization

Why Scale?

- Many ML algorithms (e.g., KNN, SVM, gradient descent-based models) are sensitive to feature magnitudes.
- Prevents dominance of one feature due to its larger scale.

Scaling Method	Formula	Output Range	When to Use
Normalization (Min-Max Scaling)	Xscaled=X−XminXmax−X minX_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}Xscaled=Xmax−X minX−Xmin	[0, 1]	When data is bounded or needed for neural networks.
Standardization (Z-score Scaling)	Xscaled=X−μσX_{scaled} = \frac{X - \mu}{\sigma}Xscaled=σX−μ	Mean = 0, SD = 1	When data has outliers or isn't bounded. Works well with SVM, logistic regression.

⚠️ Normalize when data is in known range; standardize when distributions vary or contain outliers.

---

5. Outliers: Detection & Handling

What Are Outliers?

- Data points that deviate significantly from other observations.
- Can be due to variability, errors, or rare events.

Detection Techniques:

1. **Statistical Methods**:

   - **Z-Score**:
     - Points with |z| > 3 are considered outliers.
   - **IQR Method**:
     - Outlier if x<Q1−1.5×IQR or x>Q3+1.5×IQR\text{Outlier if } x < Q1 - 1.5 \times IQR \text{ or } x > Q3 + 1.5 \times IQROutlier if x<Q1−1.5×IQR or x>Q3+1.5×IQR
   - **Grubbs' Test**:
     - Identifies one outlier at a time (assuming normal distribution).
2. **Visualization Techniques**:
   - **Boxplots**: Visually show outliers as points outside whiskers.
   - **Scatter Plots**: Useful for detecting outliers in 2D space.
   - **Histogram/Distplots**: Reveal outliers as tails or gaps.
   - **Pair Plots**: For multivariate data, reveals relationships and anomalies.
3. **Machine Learning Methods**:
   - **Isolation Forest**
   - **Local Outlier Factor (LOF)**
   - **DBSCAN** (for clustering outliers)

Handling Outliers:

- **Remove**: When confirmed as noise or error.
- **Transform**: Use log, square root, or Box-Cox transformations.
- **Cap/Floor**: Use winsorization to limit extreme values.
- **Model Robustness**: Use algorithms less sensitive to outliers (e.g., tree-based models).