KNN Imputer & Multivariate Imputation

➢ Handling Missing Data

This lecture covers **KNN Imputer**, a **multivariate** method of imputing missing data using patterns across multiple variables.

---

1. Univariate vs. Multivariate Imputation:

- ***Univariate***: Fills missing values using information from the **same column only** (e.g., mean, median).
- ***Multivariate***: Considers **multiple columns** to infer missing values — more context-aware and often more accurate.

---

2. How KNN Imputer Works:

- Finds the **k-nearest neighbors** (based on Euclidean distance) for the row with missing data.
- Uses **values from those neighbors** to estimate the missing value (mean by default).
- **Distance is calculated using available (non-missing) features only** — see scikit-learn's documentation.

---

3. Euclidean Distance with NaNs:

- The lecture explains how **distance is calculated when some features are missing**.
- scikit-learn handles this using a **masked Euclidean distance**, ignoring NaNs during the calculation.

Documentation:

- KNN Imputer - scikit-learn

---

4. Pros & Cons of KNN Imputer

❖ **Advantages**:

- Preserves **relationships between features**.
- No assumption about data distribution.

❖ **Disadvantages**:

- **Computationally expensive** with large datasets.
- **Sensitive to outliers** and irrelevant features.
- Needs **scaling** for accurate distance computation.

---

5. Uniform vs. Distance Weighting:

- weights='uniform': All neighbors contribute **equally**.

- weights='distance': **Closer neighbors** contribute more to the imputed value.

Use weights='distance' when you want **more precision** from closer data points.

➢ Key Takeaway:

KNN Imputer is a **smart multivariate technique** that uses neighbor similarity to fill in missing values. It's **more powerful than simple methods** but needs careful tuning (especially for large or unscaled data).
