Data Leakage

What is Data Leakage?

- **Definition**: When information from outside the training dataset is used to create the model, leading to overly optimistic performance estimates.
- **Result**: The model performs well during training and validation but poorly in real-world scenarios.

How Does Data Leakage Happen?

1. Split the Data Before Any Preprocessing:

🚨
Incorrect Approach (Causes Leakage):

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Load dataset
X, y = load_diabetes(return_X_y=True)

# ❌ Applying scaling before splitting (Leaks test data info)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Now split (WRONG!)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, rand
om_state=42)
```

✅
Correct Approach (No Leakage):

```
# Split data first
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat
e=42)

# Apply transformation only to training data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)  # Use same scaler, but no fit!
```

2. Feature Leakage

- Happens when a feature contains **future information** about the target variable.
- Example: A dataset predicting whether a loan will default, but including the **number of late payments** (which only happens after the loan is issued).

The **"late_payments"** column contains **post-loan** data and should NOT be included.

✅ **Fix**: Remove or create the feature only using past data.

3. Duplicate Data:

- Having duplicate samples in both training and test sets.
- Example: The same patient's data appearing in both sets in a medical dataset.

4. Hyperparameter Tuning

Detecting and Preventing Data Leakage

Review Your Features Beforehand

Unexpectedly High Performance

Inconsistent Performance Between Training and Unseen Data

Always Split Data First

- Perform **train-test split BEFORE** feature scaling or transformations.

Carefully Examine Features

- If a feature would **not be available** at prediction time, **don’t use it**.

Use Cross-Validation Properly

- If using **K-Fold CV**, ensure transformations are **done separately for each fold**.

Drop Leaking Features

- If a feature leaks information, **remove it** or **recreate it using only past data**.

Validation Set

Used for Model Tuning: The validation set allows you to adjust hyperparameters (like learning rate, number of layers, etc.) without touching the test set.

- You divide the data into 3 parts:
  1. Train → Train your data
  2. Validation → Used to tune the model.
  3. Test → Used to assess the final model's performance.

- Prevents Overfitting

Example:

If you have a dataset, you could split it into:

- **60% training data**
- **20% validation data**
- **20% test data**