Assignment 2: Understanding Data Leakage

Introduction

This assignment explores the concept of data leakage in machine learning and its impact on model performance. You'll work with a dataset related to machine predictive maintenance and identify how data leakage can occur.

Dataset

We'll be using a dataset from Kaggle related to machine predictive maintenance: https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification. This dataset aims to predict machine failures based on various sensor readings.

Assignment Parts

1. Data Preprocessing (50 points)

- **a) Load the Data:** Use pandas to load the provided "machine_data.csv" file.
- **b) Explore for Leakage:** Analyze the data to understand the features and identify potential sources of data leakage. Consider:
    - Are there features directly related to the target variable "failure" (e.g., a sensor reading indicating a critical fault)?
    - Are there categorical features that might indirectly reveal information about failure (e.g., machine type)?
- **c) Address Leakage:** Implement techniques to address data leakage:
    - Remove features that directly reveal the target label ("failure").
    - Encode categorical features (e.g., machine type) using appropriate methods (e.g., one-hot encoding).
    - Handle missing values thoughtfully (e.g., impute missing values using appropriate techniques or remove data points with too many missing values).
- **d) Document Findings:** Record your findings and the data cleaning steps you performed.

2. Model Training with and Without Leakage (50 points)

- **a) Split the Data:** Divide the preprocessed data into training, validation (optional), and test sets using a common split ratio (e.g., 80%/10%/10%).
- **b) Train Two Models:** Train two separate machine learning models (e.g., Logistic Regression, Naive Bayes) to predict machine failures using the target variable "failure."
- **c) Introduce Leakage (Optional):** For one model, introduce data leakage into the training process:
    - Use the "failure_type" feature (if it exists) as a training feature for predicting "failure." This creates leakage because failure type directly indicates if a failure occurred.
- **d) Train Both Models:** Train both models (with and without leakage) on the respective training sets.

3. Model Evaluation (50 points)

- **a) Evaluate Performance:** Assess the performance of both models on the held-out test set using appropriate metrics like accuracy, precision, recall, and F1-score.
- **b) Analyze Results:** Compare the performance of the two models and explain how data leakage impacts model performance based on your observations.

4. Data Leakage Detection Techniques (Bonus: 20 points)

- **a) Research Techniques:** Research and describe two techniques for detecting data leakage in machine learning models. These techniques could involve analyzing feature importance scores, looking for unexpected correlations, or employing specialized leakage detection algorithms.
- **b) Discuss Limitations:** Discuss the limitations or challenges associated with these detection techniques.

Deliverables

- Submit a Jupyter Notebook or Python script containing your code for data preprocessing, model training, and evaluation.
- Prepare a report summarizing your findings, including:
    - Description of identified data leakage sources.
    - Data cleaning steps performed.
    - Model performance comparison with and without leakage.
    - Explanation of data leakage impact on model performance.
    - Description of data leakage detection techniques (bonus).

Evaluation Criteria

- Correctness and completeness of data preprocessing steps (20 points).
- Implementation and training of models with and without leakage (20 points).
- Evaluation and analysis of model performance (20 points).
- Understanding and explanation of data leakage impact (20 points).
- Research and discussion of data leakage detection techniques (bonus: 20 points).