Lab 3 Assignment

Data Preprocessing with heart Dataset

Objectives

1. Standardize numerical features to have mean = 0 and std = 1.
2. Apply one-hot encoding on categorical variables.
3. Ensure no data leakage by fitting scalers only on training data.

Tools

- Scikit-learn
- Pandas

Dataset (free & open)

Heart Disease UCI Dataset (Kaggle)
This dataset has both numerical and categorical features, making it perfect for this exercise.

Supporting Content

- **Standardization (Z-score normalization):** Needed for algorithms like logistic regression, k-NN, and SVM.
- **One-Hot Encoding:** Converts categorical columns (like cp, thal) into binary features.
- **Data Leakage:** Never fit scalers/encoders on the full dataset before splitting.

Lab Tasks

1. **Separate features and target**
    - Target variable: target (1 = disease, 0 = no disease).
2. **Split the data**
3. **Standardize numerical features**
    - Columns: age, trestbps, chol, thalach, oldpeak.
    - Use StandardScaler, fit on training only, transform both train and test.
4. **One-Hot Encode categorical features**
    - Columns: cp, thal, slope.

- Use pd.get_dummies(..., drop_first=True).

5. **Check processed training data**
    - Inspect first few rows after preprocessing.

Exercises

1. What are the mean and std of the standardized chol column in the training set? Why aren't they exactly 0 and 1 when checked with pandas.Series.std()?
2. Why do we use drop_first=True in one-hot encoding? What problem does it solve?
3. If a new category of thal appeared in the test set but not in the training set, what would happen with pd.get_dummies? How can OneHotEncoder(handle_unknown='ignore') in scikit-learn help here?