Data Mining Assignment Report

This report provides brief answers to the questions in the data mining assignment 1.

# Exercise 1: Loading Data with Pandas

1. Loading a CSV file into a Pandas DataFrame can be done using the `pd.read\_csv()` function. It takes the file path as an argument and returns the dataset as a DataFrame.

2. The `info()` function provides information about the DataFrame such as column names, data types, number of non-null values, and memory usage.

3. Missing values can be identified using the `isnull()` function combined with `sum()` to count them.

# Exercise 2: Handling Missing Data

1. Various strategies can be used to handle missing data. In this case, the `dropna()` function can be used to remove rows with missing values or the `fillna()` function can fill missing values using the mean, median, or a specific value.

2. Filling missing values can alter the dataset's distribution, especially for numeric features, depending on the method chosen.

3. Dropping rows may be preferred when missing data is minimal and filling could introduce bias.

# Exercise 3: Data Transformation

1. Normalization scales the values to a specific range (e.g., 0 to 1), while standardization scales the values based on the mean and standard deviation.

2. One-hot encoding converts categorical variables into binary columns, making them compatible with machine learning models.

3. Binning continuous variables can be useful when the relationship between the feature and target is non-linear or to reduce noise.

# Exercise 4: Feature Engineering

1. New features can be created by interacting or transforming existing features. For example, interaction terms between two numerical features or polynomial features can be added to improve predictive power.

2. Date-based features, such as extracting the year, month, or day from datetime columns, can capture temporal patterns in the data.

# Exercise 5: Data Cleaning

1. Duplicate rows can be removed using the `drop\_duplicates()` function.

2. Outliers can be detected and removed using the Z-score method (by removing points outside 3 standard deviations) or the IQR method (by removing points outside the 1.5 \* IQR range).

3. Inconsistent categorical data can be standardized using string methods (e.g., `str.lower()` for case standardization).

# Exercise 6: Splitting Data into Training and Testing Sets

1. The `train\_test\_split()` function from the sklearn library is used to split the dataset into training and testing sets.

2. It is essential to ensure that the target variable is kept separate from the features during the split.

3. The size of the training set affects the model's ability to generalize: too small and the model may underfit, too large and it may overfit.

# Exercise 7: Data Preprocessing Pipeline

1. A preprocessing pipeline ensures that the data preparation steps, such as missing value imputation and feature scaling, are applied consistently.

2. The pipeline ensures that the same transformations are applied to both the training and test sets.

3. Additional steps, such as feature engineering or dimensionality reduction, can be added to the pipeline for automation.