**Assignment 1, Data Mining**

Put all deliverables into github repository in your profile. Share link to google form 24 hours before defense. Defend by explaining deliverables and answering questions.

Deliverables: .ipynb

Google form: <https://docs.google.com/forms/d/e/1FAIpQLSe0GyNdOYlvM1tX_I_CtlPod5jBf-ACLGdHYZq1gVZbUeBzIg/viewform?usp=sf_link>

### **Exercise 1: Loading Data with Pandas**

1. **Objective**: Learn how to load and inspect datasets using Pandas.
2. **Steps**:
   * Import the Pandas library and load a CSV file into a DataFrame.
   * Use the head(), tail(), and info() functions to inspect the dataset.
   * Check for missing values and data types of each column using isnull() and dtypes.
3. **Questions**:
   * How do you load a CSV file into a Pandas DataFrame?
   * What information does the info() function provide about the dataset?
   * How can you identify missing values in the dataset?

### **Exercise 2: Handling Missing Data**

1. **Objective**: Practice techniques for handling missing data in a dataset.
2. **Steps**:
   * Identify missing values in the dataset using isnull().sum().
   * Use different strategies to handle missing data:
     + Remove rows with missing values using dropna().
     + Fill missing values with the mean, median, or a specific value using fillna().
     + Use forward or backward filling (ffill() or bfill()) to fill missing data.
   * Compare the results of each method.
3. **Questions**:
   * What strategy did you use to handle missing values, and why?
   * How did filling missing values affect the dataset?
   * When might it be more appropriate to drop rows with missing values instead of filling them?

### **Exercise 3: Data Transformation**

1. **Objective**: Transform data to prepare it for analysis.
2. **Steps**:
   * Normalize numerical features using Min-Max scaling or Z-score standardization with sklearn.preprocessing.
   * Encode categorical variables using one-hot encoding with pd.get\_dummies() or sklearn.preprocessing.OneHotEncoder.
   * Use pd.cut() to bin continuous variables into discrete intervals.
3. **Questions**:
   * What is the difference between normalization and standardization?
   * How does one-hot encoding transform categorical variables?
   * Why might you want to bin continuous variables into categories?

### **Exercise 4: Feature Engineering**

1. **Objective**: Create new features to improve the predictive power of a dataset.
2. **Steps**:
   * Create new features by combining or transforming existing features (e.g., adding interaction terms or polynomial features).
   * Extract date-based features (e.g., year, month, day) from datetime columns using pd.to\_datetime() and dt accessor.
   * Use domain knowledge to engineer features that might be useful for your specific problem.
3. **Questions**:
   * What new features did you create, and why?
   * How did the new features improve the dataset?
   * How can date-based features be useful in a dataset?

### **Exercise 5: Data Cleaning**

1. **Objective**: Clean data to ensure it's ready for analysis.
2. **Steps**:
   * Remove duplicate rows using drop\_duplicates().
   * Detect and remove outliers using the Z-score method or the IQR method.
   * Correct inconsistencies in categorical data (e.g., standardizing text formats or merging similar categories).
3. **Questions**:
   * How did you identify and handle duplicate rows in the dataset?
   * What method did you use to detect and remove outliers, and why?
   * How did you address inconsistencies in categorical data?

### **Exercise 6: Splitting Data into Training and Testing Sets**

1. **Objective**: Prepare the data for model training by splitting it into training and testing sets.
2. **Steps**:
   * Use sklearn.model\_selection.train\_test\_split() to split the dataset into training and testing sets.
   * Ensure that the target variable is correctly separated from the features.
   * Explore the impact of different train-test split ratios (e.g., 70-30, 80-20) on model performance.
3. **Questions**:
   * How do you split a dataset into training and testing sets in Python?
   * What considerations should you keep in mind when choosing a train-test split ratio?
   * How does the size of the training set impact the model's ability to generalize?

### **Exercise 7: Data Preprocessing Pipeline**

1. **Objective**: Build a preprocessing pipeline to automate the data preparation process.
2. **Steps**:
   * Use sklearn.pipeline.Pipeline to create a pipeline that includes steps such as missing value imputation, feature scaling, and encoding categorical variables.
   * Fit the pipeline to the training data and transform the test data.
   * Integrate the preprocessing pipeline with a machine learning model for end-to-end training and evaluation.
3. **Questions**:
   * What are the benefits of using a preprocessing pipeline?
   * How does the pipeline ensure consistency between training and test data transformations?
   * How can you extend the pipeline to include additional preprocessing steps?