Assessment Guidelines for **ILO 5.0: Data Understanding and Preparation**

Poor

1- The individual contributions of team members are documented.
2- The necessary code to connect to SQL database for the project is provided.

Insufficient

1- A simple query is provided showing that the student has the ability to access data in the SQL database for the project.

Sufficient

This criterion is related to simple preprocessing steps as follows:

1- Remove unnecessary columns from the dataset
2- Handling missing values by:
    a. Removing Missing Values: remove rows or columns that contain missing values
    b. Filling Missing values: fill missing values with a specific values, such as the mean, median, mode, or a fixed value
    c. Interpolation: use interpolation methods to estimate and fill missing values
    d. Flagging Missing value: Create an additional column to indicate the presence of missing values
3- Handling outliers:
    a. Identify outliers (e.g. using visualization, using statistical method like z-score
    b. Remove outliers
    c. Transforming data: Treating outliers by transforming data e.g. using log transformation, square root transformation, imputing outliers with the mean or median of the column
4- Convert categorical variables to numerical variables e.g. by using one-hot encoding
5- Split the dataset into training, validation, and test sets

Not all of the above steps may be relevant for every project, but the team should at least provide evidence for the following tasks:

1- Evaluate whether there are missing values in the data planned for use. If so, the team needs to handle them.

2- Check for the presence of outliers. If any are found, the team needs to address them.
3- Split the dataset into **training**, **validation**, and **test** sets. Alternatively, if there isn't enough data, split it into **training** and **test** sets.

Good

The potential tasks that could be considered for this criterion are as follows:

1- Providing the code for joining datasets (using a JOIN in SQL)
2- Normalizing features using one or more of the following methods:
    a. Min-Max Scaling (Using MinMaxScaler from scikit-learn)
    b. Standardization (Using StandardScaler from scikit-learn)
    c. Max Absolute scaling (Using MaxAbsScaler from scikit-learn)
    d. Robust Scaling (Using RobustScaler from scikit-learn)
3- Creating new features using one of the following approaches:
    a. Combine existing features
    b. Extract information from dates
    c. Aggregating data (especially for time series data or for grouped data)
    d. Create interaction features (by production of two or more features to capture relationship between them)
    e. Creating polynomial features

Note that all these steps are not applicable for all the projects. But the team needs to show evidence for at least the following:

1- Normalizing features (essential for regression and classification problems)
2- Using JOIN in SQL if they use more than one dataset (if applicable)
3- Create features like day, week, month, and year features from the date and time information (if applicable)
4- Aggregate the data if the team group them by day, month, week, …(if applicable)

Excellent

The potential tasks that could be considered for this criterion are as follows:

1- Using SQL to create VIEW from data
2- In a classification problem, analyse the distribution of output classes, for example, by using histogram visualizations or value counts to understand the class distribution.
3- If an imbalance in the number of instances belonging to each class is identified, use one of the following techniques:

   a. Random oversampling (randomly duplicates samples from the minority class until the class distribution is balanced; e.g. using RandomOverSampler from imblearn.over_sampling)
   b. Random Undersampling (Randomly removes samples from the majority class until the class distribution is balanced, e.g. using RandomUnderSampler from imblearn.under_sampling)
   c. Random Sampling with Replacement (Randomly sampling with replacement from the majority class to match the number of samples in the minority class)
   d. Use SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic samples in the feature space of the minority class to balance the distribution; e.g. using SMOTE from imblearn.over_sampling)
   e. Use class weighting to penalize misclassification of minority classes more than majority classes during training (in most of the classification there is a hyperparameter called `class_weight` that can be used.)

4- Identifying the skewness in the output variable (crucial for building accurate models in regression problems). Plot a histogram of the output variable to gain insight into its distribution. To address the skewness, one of the following methods can be used:
   a. Log transformation
   b. Square root transformation
   c. Box-cox transformation

5- Document the preprocessing steps applied to the data in the form of a PDF file.

Note that all these steps may not be applicable for all projects; however, the team needs to provide the following evidence to meet this criterion:

1- Using SQL to create VIEW from data.
2- For classification problems, analyse the distribution of the target classes. If an imbalance is identified, it is suggested to be addressed.
3- For regression problems, analyse the skewness of the output variable. If skewness is identified, it is suggested to be addressed.
4- Provide a PDF file detailing all preprocessing steps.