CCA-5: Ensure data integrity and consistency

Overview

It is very important to check data validity before analysis to ensure that the data which will be used in the analysis is proper and error-free. Data validation helps the analysts to know about errors, inconsistencies, or anomalies which could affect quality analysis. Please document what validation steps you took and the results of these validation steps in the process of finalizing the dataset for this project: it is also helpful if you give a brief note for each step on why you used a given method, or what you learned about the data from your result for each step.

Objectives

- To ensure that the data is free from any errors and is complete.
- To identify and handle the anomalies or inconsistencies that might affect the analysis.
- To ensure that the data transformations and feature engineering steps have been applied correctly.

Steps

1. **Initial Data Integrity Checks**
   - **Objective**: Ensure that the data is complete and free from obvious errors.
   - **Actions**:
     - **Missing Values**: Checked for missing values in the dataset.
     - **Data Types**: Ensured correct data types of all columns.
     - **Duplicated Records**: Identified 103 cases where records are duplicated within the dataset. After sufficient discussion and consideration with the team, we decided not to remove this set of duplicated records at this point. Given the timeline and possible impact on the integrity of the dataset, the removal of these records might add unnecessary complexity and risk to the analysis, especially at the stage we are at nearing task submission. Further, these duplicates are retained to allow for more in-depth analysis as all variations possible in customer behavior can be captured.
   - **Code Snippet**:

```
# Checking for missing values
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)

# Check data types
data_types = df.dtypes
print("Data Types:\n", data_types)

# Checking for duplicates
```

```
duplicate_count = df.duplicated().sum()
print(f"No of Duplicates: {duplicate_count}")

```

2. **Consistency and Range Checks**
   - **Task**: To validate the consistency and range of the data values.
   - **Steps Taken**:
     - **Outliers**: Checked for outliers that can significantly affect the analysis.
     - **Range Checks**: Checked that the numerical features fall between the expected ranges.
     - **Category Validation**: Checked that categorical features have valid categories only.
   - **Code Snippet**:

```
# Find outliers using IQR
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
outliers = ((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).sum()
print("Outliers found:\n', outliers)

# Inspect the range of numerical columns
for column in numerical_columns:
    print(f"{column} range: {df[column].min()} to {df[column].max()}")

# Check for unexpected categories
for column in categorical_columns:
    unique_categories = df[column].unique()
    print(f"Categories in {column}:\n", unique_categories)

```

3. **Transform Validation**
   - **Objective**: To make sure the data transformation has been applied correctly post feature engineering.
   - **Steps Performed**:
     - **Scale Validation**: Checked if we had applied feature scaling (Min-Max, Standard) perfectly.
     - **Encoding Validation**: Checked if we had applied the categorical encoding perfectly.
   - **Code Snippet**:

```
# Checking scaling on one column
original_min = df_original['MonthlyCharges'].min()
original_max = df_original['MonthlyCharges'].max()
scaled_min = df_scaled['MonthlyCharges'].min()
scaled_max = df_scaled['MonthlyCharges'].max()
print(f"Original range: {original_min} to {original_max}")
print(f"Scaled range: {scaled_min} to {scaled_max}")

# Checking if the encoding is consistent
encoded_columns = df_encoded.columns
print("Encoded Columns :\n", encoded_columns)

```

Importance of Data Validity

- **Data Accuracy**: It allows the analysis to be based on correct and reliable data, hence arriving at dependable insights.

- **Model Performance**: Accurate data makes the model more correct and reduces the chances of inaccuracy in predictions.
- **Compliance**: Maintains the standards of data integrity and keeps the project within the specifications provided for data quality.

Results Obtained

- **Data Integrity**: This has been confirmed, and it is noted that there is no missing value in the dataset, all entries in all features are valid, and duplication of 103 records is retained, assuming a conscious decision to maintain the complexity of the data.
- **Outlier Management**: Detected potential outliers in the data that will lead to distortion in analysis if not treated.
- **Transformation Verification**: Checks whether the data has undergone transformations such as scaling and encoding.

Next Steps

- **Cluster Analyzing**: Now that our data is validated, we can now cluster customer segmentation.
- **Feature Engineering Review**: If something is wrong during clustering, then revisit feature engineering to make sure all the features contribute positively to the analysis.

Conclusion

Data validity is one of the critical processes that ensure accuracy and dependability in our analysis. The decision to keep duplicates was made so that we do not lose richness in data and miss any kind of changing behavior. Now the validated data is ready for the following phase of clustering analysis.