Report on `task1.ipynb`

Objective: The primary objective of this notebook is to analyze and preprocess textual data related to customer and correction verbatim, likely to extract meaningful insights or prepare it for further analysis.

Data Loading and Inspection:

- The script begins by importing necessary libraries: `pandas`, `numpy`, `re`, `nltk` (including `stopwords` and `word_tokenize`), `seaborn`, `plotly.express`, `matplotlib.pyplot`, and `collections.Counter`. These libraries are standard for data manipulation, cleaning, text processing, visualization, and counting occurrences.
- It loads data from an Excel file named "Data for Task 1.xlsx" into a pandas DataFrame named `df`.
- The `df.info()` method is used to display the structure of the DataFrame, including column names, data types, and non-null counts. This step is crucial for understanding the dataset's characteristics and identifying potential data quality issues (e.g., missing values).
- The `df.shape` method reveals the dimensions of the DataFrame (number of rows and columns), giving an overview of the data size.
- `df.head(5)` displays the first 5 rows, allowing for a quick inspection of the data content and format.
- `pd.set_option('display.max_columns', None)` ensures that all columns of the DataFrame are displayed, which is useful for a comprehensive view of the data.
- `df.describe()` provides descriptive statistics for numerical columns, giving insights into the distribution and range of values.
- `df.isnull().sum()` calculates the number of missing values in each column, highlighting where data cleaning may be needed.

Text Preprocessing and Analysis:

- A function `clean_text` is defined to preprocess text data. This function performs the following steps:
    - It converts the text to lowercase.
    - It removes any characters that are not alphanumeric or whitespace using regular expressions.
    - It tokenizes the text into individual words.
    - It removes stopwords (common words like "the," "is," "in") using `nltk.corpus.stopwords`.
    - It filters out words with a length of 2 or fewer characters.

- The `clean_text` function is applied to a combined text field (`df['combined_text']`), which is created by concatenating the `CORRECTION_VERBATIM` and `CUSTOMER_VERBATIM` columns. The resulting tokens are stored in a new column called `tokens`.
- All tokens are flattened into a single list `all_tokens`.
- The `Counter` class is used to count the frequency of each token, and `most_common(30)` retrieves the 30 most frequent tokens.
- The top 30 tokens and their frequencies are displayed in a DataFrame `tags_df`.

Output:

- The final DataFrame `df`, which now includes the `tokens` column, is saved to a CSV file named "task1.csv" using `df.to_csv()`.

Summary:

The `task1.ipynb` notebook performs a standard text analysis workflow: loading data, inspecting its basic properties, preprocessing the text by cleaning and tokenizing, and then analyzing word frequencies. The output is a cleaned dataset with tokenized text and a frequency count of the most common words, which can be used for further analysis like topic modeling, sentiment analysis, or text classification.