Python Assignment

Excel Data Manipulation

Task 1: Read and Summarize Data

1. Read an Excel file (data.xlsx) into a DataFrame using **pandas**.
2. Print the first 10 rows of the dataset.
3. Display the number of rows and columns in the dataset.
4. Check for missing values in each column and display the count.

Expected Output:

- First 10 rows of the dataset.
- Dataset shape and missing value summary.

Task 2: Add a New Column

1. Read an Excel file (sales_data.xlsx).
2. Add a new column that calculates a **percentage change** between two columns (e.g., Sales and Cost).
3. Save the updated data to a new file (updated_sales_data.xlsx).

Expected Output:

- A new Excel file with the added column.

Task 3: Filter Rows

1. Read an Excel file (employees.xlsx).
2. Filter out rows where Salary is greater than 50,000.
3. Save the filtered data into a new Excel file (high_salary_employees.xlsx).

Expected Output:

- A new Excel file containing only the filtered rows.

Task 4: Handle Missing Data

1. Read an Excel file (product_data.xlsx).
2. Replace missing values in numerical columns with their mean.
3. Replace missing values in categorical columns with their mode.
4. Save the cleaned data into a new file (cleaned_product_data.xlsx).

Expected Output:

- A new Excel file with missing values handled appropriately.

Task 5: Combine Multiple Excel Files

1. Assume there are multiple Excel files in a folder (/data_files/).
2. Write a script to read all files, combine them into a single DataFrame, and save the combined data into combined_data.xlsx.

Expected Output:

- A new Excel file containing data from all input files.

Data Cleaning and Transformation

Task 6: Normalize Data

1. Read a dataset (normalization_data.xlsx) with numerical columns.
2. Normalize all numerical columns using **Min-Max Scaling**.
3. Save the normalized data into normalized_data.xlsx.

Expected Output:

- A new Excel file with normalized data.

Task 7: Encode Categorical Data

1. Read a dataset (categorical_data.xlsx) with a column named Category.
2. Convert the Category column to numerical values using **Label Encoding**.
3. Save the encoded dataset into encoded_data.xlsx.

Expected Output:

- A new Excel file with the Category column encoded.

Task 8: Split Data for Training

1. Read a dataset (training_data.xlsx) with features (Feature1, Feature2) and a target column (Target).
2. Split the dataset into **training** (70%) and **testing** (30%) sets using **train_test_split**.
3. Display the size of training and testing sets.

Expected Output:

- Number of rows in training and testing sets.

Text Data Handling

Task 9: Clean Text Data

1. Read a dataset (reviews.xlsx) with a column Review_Text.
2. Write a script to:
    - Remove all special characters.
    - Convert text to lowercase.
3. Save the cleaned text data into cleaned_reviews.xlsx.

Expected Output:

- A new Excel file with cleaned text.

Task 10: Count Word Frequency

1. Using the cleaned_reviews.xlsx file, count the frequency of each word in the Review_Text column.
2. Display the top 10 most frequent words and their counts.

Expected Output:

- A dictionary of the top 10 words and their counts.

Data Visualization

Task 11: Create a Bar Chart

1. Read a dataset (sales_data.xlsx) with a column Region.
2. Create a bar chart showing the total sales per region.

Expected Output:

- A bar chart displaying total sales per region.

Task 12: Create a Correlation Heatmap

1. Read a dataset (data.xlsx) with numerical columns.
2. Compute the correlation matrix of the dataset.
3. Plot a heatmap of the correlation matrix.

Expected Output:

- A heatmap visualizing correlations.

Foundational AI-Related Tasks

Task 13: Simple Linear Regression

1. Read a dataset (housing_data.xlsx) with columns SquareFeet (input feature) and Price (target).
2. Train a **Linear Regression** model to predict Price based on SquareFeet.
3. Print the model's coefficients.

Expected Output:

- Model's coefficients.

Task 14: Prepare Data for ML

1. Read a dataset (ml_data.xlsx) with both categorical and numerical columns.
2. Perform the following:
    - Normalize numerical columns.
    - Encode categorical columns.
3. Save the prepared data into prepared_data.xlsx.

Expected Output:

- A new Excel file with normalized and encoded data.

API Integration

Task 15: Fetch Data from an API

1. Use the **requests** library to fetch JSON data from https://jsonplaceholder.typicode.com/posts.
2. Save the data into an Excel file (api_data.xlsx).

Expected Output:

- An Excel file containing the fetched data.

Task 16: Automate API Data Processing

1. Write a script to:
    - Fetch JSON data from an API.
    - Clean the data (e.g., remove unnecessary fields).
    - Save the cleaned data into processed_api_data.xlsx.

Expected Output:

- A new Excel file with cleaned data.

Automation with Error Handling

Task 17: Robust Excel Handling

1. Write a script that:
    - Reads an Excel file (data.xlsx).
    - Handles errors like FileNotFoundError or missing columns.
    - Logs the error and exits gracefully if something goes wrong.

Expected Output:

- A robust script with error-handling mechanisms.