Task 1: Data Cleaning and Preprocessing
Report Task 1
Elevate Labs

1. Dataset Used

- **Dataset:** Medical Appointment No Shows
- **Source:** Kaggle - https://www.kaggle.com/datasets/joniarroba/noshowappointments
- **Rows/Columns:** 110,527 rows and 14 columns
- **Objective:** Clean and preprocess the raw dataset to prepare it for analysis.

2. Data Cleaning and Preprocessing Steps

- Loaded the dataset using Pandas.
- Checked for missing values using .isnull().sum().
- Removed duplicate rows using .drop_duplicates().
- Standardized 'Gender' and 'No-show' columns to uppercase using .str.upper().
- Converted 'ScheduledDay' and 'AppointmentDay' columns to datetime format using pd.to_datetime().
- Renamed all column headers to lowercase with underscores for consistency.
- Fixed data types (e.g., age, scholarship, diabetes to int).
- Removed rows with negative age values.
- Exported the cleaned data to 'cleaned_noshowappointments.csv'.

3. Python Code Used

```
import pandas as pd

# Load the dataset
df = pd.read_csv('noshowappointments.csv')

# Display initial information
print("Initial Data Info:")
print(df.info())
print("\nMissing Values:")
print(df.isnull().sum())

# Remove duplicate rows
df.drop_duplicates(inplace=True)

# Handle missing values (if any)
# For this dataset, there are no missing values, but this is a general approach
# df.fillna(method='ffill', inplace=True)

# Standardize text values
df['Gender'] = df['Gender'].str.upper()
df['No-show'] = df['No-show'].str.upper()

# Convert date columns to datetime
df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'])
```

```
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'])

# Rename column headers to be lowercase with underscores
df.columns = [col.strip().lower().replace('-', '_') for col in df.columns]

# Fix data types
df['age'] = df['age'].astype(int)
df['scholarship'] = df['scholarship'].astype(int)
df['hipertension'] = df['hipertension'].astype(int)
df['diabetes'] = df['diabetes'].astype(int)
df['alcoholism'] = df['alcoholism'].astype(int)
df['handcap'] = df['handcap'].astype(int)
df['sms_received'] = df['sms_received'].astype(int)

# Handle inconsistent data
# Remove rows with negative age
df = df[df['age'] >= 0]

# Save the cleaned dataset
df.to_csv('cleaned_noshowappointments.csv', index=False)

print("\nData cleaning complete. Cleaned dataset saved as 'cleaned_noshowappointments.csv'.")
```

4. <u>Summary of Changes</u>

- No missing values were found
- Removed 100+ duplicate rows
- Text values like Gender and No-show were standardized.
- Date columns were converted to dd-mm-yyyy format.
- Column headers were renamed to clean and uniform format.
- Negative age entries were removed.
- Final cleaned dataset was saved as 'cleaned_noshowappointments.csv'.

<u>Name:</u> Akash Kumar Rajak
<u>Role:</u> Data Analyst Intern
<u>Internship Company:</u> Elevate Labs
<u>Date of Submission:</u> 02-06-2025
<u>Email:</u> akashkumarrajak200@gmail.com
<u>Contact No.</u> 9711671664