Data Cleaning

Overview

This Python script is designed to clean and preprocess a dataset containing information about individuals, such as booking details, personal attributes, and legal information. The goal is to standardize and enhance the data for further analysis.

Usage

1. Initialization:

- Create an instance of the DataCleaning class by providing the dataset as a parameter.

```
data = pd.DataFrame(your_data)  # Replace 'your_data' with your actual dataset
```

```
data_mapper = DataCleaning(data)
```

2. Column Mapping:

- The script performs column renaming based on a predefined mapping to ensure consistency and standardization.

This function, named rename_columns, is designed to rename columns in a DataFrame, filter the DataFrame to keep only specified columns, and reindex the DataFrame to ensure that all columns, including those not present in the original DataFrame, are included with missing values set to NaN.

```
data_mapper.rename_columns()
```

3. Gender Mapping:

- Maps gender-related columns to a standardized format.

The map_gender function is designed to preprocess the 'sex' column in a DataFrame by filling missing values with 'Unknown' and mapping the existing values using a predefined gender_mapping dictionary

```
data_mapper.map_gender()
```

4.Race Mapping:

- Maps race-related columns to a standardized format.

The map_race function is designed to preprocess the 'race' column in a DataFrame. It involves filling missing values with 'Unknown' and mapping the existing values using a predefined race_mapping dictionary

```
data_mapper.map_race()
```

5. Eye Color Mapping:

- Maps eye color-related columns to a standardized format.

The map_eyColor function is designed to preprocess the 'eyeColor' column in a DataFrame. It involves filling missing values with 'Unknown' and mapping the existing values using a predefined eyeColor_mapping dictionary

```
data_mapper.map_eye_color()
```

6. Hair Color Mapping:

- Maps hair color-related columns to a standardized format.

The map_hairColor function is designed to preprocess the 'hairColor' column in a DataFrame. It involves filling missing values with 'Unknown' and mapping the existing values using a predefined hairColor_mapping dictionary

```
data_mapper.map_hair_color()
```

7. Housing Mapping:

- Maps housing-related columns to a standardized format. (Note: The 'housing1' mapping is currently commented out in the script.)

The function is designed to preprocess the 'housing1' column in a DataFrame. It involves filling missing values with 'Unknown' and mapping the existing values using a predefined county_facilities, El_Dorado_Housing and Placer_Housing dictionary

This function is only applied for:

- Amador
- Calaveras
- Humboldt
- Inyo
- Kern
- Lake
- Kings
- Madera
- Marin
- Mendocino
- Mono
- Napa
- Nevada
- San Luis Obispo

- Santa Cruz
- Shasta
- Sutter
- Tehama
- Yuba
- El Dorado
- Placer
- Yuba

```
data_mapper.map_housing1()
```

8. Top Charge Mapping:

- Maps top charge-related columns to a standardized format.

The function is designed to preprocess the 'topcharge' column in a DataFrame. It involves filling missing values with 'Unknown' and mapping the existing values using a predefined topcharge_mapping dictionary

```
data_mapper.map_topcharge()
```

9. Height Conversion:

- Converts height entries to inches for uniformity.

The convert_height function accommodates different scenarios in raw data representations of heights, such as "5'04" or "5 feet 4" (there are several different cases). The main conversion is handled by the nested height_to_inches function, which intelligently splits the string, extracts feet and inches, and calculates the height in inches. The function is designed to be robust, providing a standardized representation for various height formats found in the raw data.

```
data_mapper.convert_height()
```

10. Weight Cleaning:

- Cleans and converts weight entries to numeric format.

The clean_weight function standardizes the 'weight' column in a DataFrame by replacing 'Unknown' with a default weight (183.5), removing 'Lbs' or 'Lbs.' strings, and converting the column to numeric values with error handling for non-convertible values.

```
data_mapper.clean_weight()
```

11. Bail and Bond Cleaning:

- Cleans and converts bail and bond entries to numeric format.

The clean_bail_and_bond function standardizes the 'bail' column in a DataFrame by converting it to a string, removing '$' and commas, and converting it to numeric values with error handling for non-convertible values.

```
data_mapper.clean_bail_and_bond()
```

12. Date Conversion:

- Converts date-related columns to the 'YYYY-MM-DD' format.

The code snippet processes specified date columns in a DataFrame by replacing 'Unknown' with NaT, converting the columns to datetime objects, and formatting them to 'YYYY-MM-DD'

```
data_mapper.convert_dates()
```

13. Age Calculation:

- Calculates age based on the provided date of birth and booking date.

The calculate_age function computes the age of individuals in a DataFrame using the 'date' and 'dob' columns. It calculates the age by subtracting the birth year from the current year. If the 'dob' column is missing, it fills the 'age' column with the mean age, excluding 'Unknown' values. This function is applicable when the data lacks an 'age' column but includes 'dob'

```
data_mapper.calculate_age()
```

14. Incarcerated Days Calculation:

- Calculates the number of days an individual has been incarcerated.

The calculate_incarcerated_days function computes the duration of incarceration ('incarcerated_days') in a DataFrame by subtracting the 'firstappearance' date from the 'date' date. Both 'firstappearance' and 'date' columns are ensured to be in datetime forma

```
data_mapper.calculate_incarcerated_days()
```

15. Data Display:

- Displays unique values in mapped columns and shows the cleaned dataset.

```
data_mapper.display_data()
```

Data Output

- The processed and cleaned dataset is stored in the data_mapper.data attribute.

```
df = data_mapper.data
```

Notes

- The script includes extensive mapping dictionaries for gender, race, eye color, hair color, housing, and top charges. Ensure these mappings align with your dataset's unique values.

- Adjust the county_facilities, El_Dorado_Housing, and Placer_Housing dictionaries as needed for accurate housing information.
- The script makes assumptions about certain column entries and may need further adjustments based on your specific dataset.

Dependencies

- This script relies on the pandas library for data manipulation.

```
import pandas as pd
```