Documentation of Booking Scraper Application

Julia Bednarczyk
Student of AGH UST
Computer Science and Econometrics
GitHub Profile

December 2024

About this Application

Welcome to the **Booking Scraper**! This application combines **web scraping**, **interactive visualizations**, and **analysis** to simplify travel planning.

What can you do?

- **Search Hotels**: Input a destination, travel dates, and the number of guests to explore options.
- **Scrape Data**: Collect detailed information about hotels, including prices, reviews, and distances.
- **Interactive Map**: Visualize hotel locations and highlight the top 5 hotels based on your criteria.
- **Analyze Trends**: Explore scatter plots, 3D visualizations, and histograms to understand market trends.
- **Customize Your Search**: Refine results by selecting key metrics such as price, ratings, or proximity.

How to use the App?

1. **Start with the Home Page**:
    - Enter the city name, check-in and check-out dates, and the number of adults.
    - Click *Find My Stay* to scrape data from Booking.com.
2. **Explore Hotel Information**:
    - Navigate through tabs to view detailed stats, interactive maps, and customizable visualizations.
3. **Understand Trends**:
    - Use scatter plots and histograms to analyze relationships like price vs reviews.
4. **Top Picks**:
    - Identify the best hotels based on your preferred metrics (e.g., lowest price, highest rating).
    - **Pro Tip**: If you're unsure about any feature, hover over the question mark icons to get more details.

What's under the hood?

The application is available on GitHub: https://github.com/pydaisy/BookingScraperApp
The application leverages modern Python libraries and frameworks:

- **Scrapy**: For scraping hotel data from Booking.com.
- **Streamlit**: To create an interactive and user-friendly interface.
- **Pandas**: For data preprocessing and analysis.
- **Plotly**: For dynamic 2D and 3D visualizations.
- **Folium**: For interactive hotel location maps.
- **Custom Design**: Includes light/dark mode with a unique color palette.

Code Structure

The application is modularly organized:

- `scraper_booking.py`: Handles scraping hotel lists and essential data.
- `scraper_hotel_details.py`: Extracts additional information like coordinates.
- `app.py`: Manages application logic, user interface, and data presentation.
- `theme_settings.py`: Configures light and dark modes.

Main Functions in the Modules

`scraper_booking.py`

- `__init__(self, url)`: Generates URLs for scraping based on user preferences.
- `parse(self, response)`: Extracts hotel details such as name, address, price, and reviews.
- `close(self, reason)`: Removes duplicate entries and triggers additional scraping.
- `remove_duplicates_from_csv(file_path)`: Cleans the dataset by removing duplicates.
- `run_spider(url)`: Starts the scraping process.

Key Features

- Supports sorting results in various ways (e.g., price, distance, ratings).
- Automatic removal of duplicates.
- Dynamically configurable with any provided URL.

app.py

- `create_map(scraped_data, top_5_hotels=None, filter_top_5=False)`: Generates interactive maps.
- `load_scraped_data(file_path)`: Prepares data for analysis and visualization.
- `home_content(dark_mode)`: Displays the app's main interface and starts the scraping process.
- `run_scraper(url)`: Manages the data collection process.

Key Features

- **Interactive Map:** Visualize hotel locations on an interactive map with markers that provide additional details, such as price range and rating.
- **Hotel Search:** Users can search for hotels by entering travel details such as city, dates, and guest count. The app then scrapes relevant hotel information from Booking.com.
- **Data Insights:** After the scraping process is complete, the app displays key statistics such as average price, review score, and number of hotels available.
- **Customizable Filters:** Users can filter hotels by specific criteria, such as top-rated, most affordable, or closest to the city center.
- **Dark Mode:** The app supports dark mode for improved user experience in low-light environments.
- **Error Handling:** The app is built with robust error handling, ensuring smooth user interaction even in cases of incomplete or invalid input.
- **Price Range Classification:** Hotels are classified into different price ranges, allowing users to easily find options within their budget.
- **Booking Link:** After the search, the app generates a direct link to Booking.com for users to make reservations.

This Streamlit app is designed to be both informative and user-friendly, providing a streamlined process for discovering hotels and analyzing key data points in a visually appealing way. With the integration of interactive maps and dynamic filtering, it enhances the overall travel planning experience for users.

`scraper_hotel_details.py`

- `__init__(self, csv_file)`: Loads hotel links for detailed scraping.
- `parse(self, response)`: Extracts hotel type and coordinates.
- `close(self, reason)`: Cleans the dataset after scraping.
- `remove_duplicates_from_csv(file_path)`: Ensures unique entries in the dataset.

Key Features

- Dynamically scrapes hotel pages using start URLs.
- Ensures data accuracy with automatic duplicate removal.
- Seamlessly integrates with the `scraper_booking.py` script for enriched datasets.

`theme_settings.py`

The `theme_settings.py` file is responsible for managing the theme and color scheme of the Streamlit application. It allows users to apply custom themes, including light and dark modes, and generates a color palette based on the selected theme for a more visually appealing user interface.

Main functions:

- `load_theme(file_path)` Loads a theme from a JSON file, ensuring that the app can dynamically change its appearance based on the theme settings.
- `apply_theme(dark_mode, theme_file)` Applies the selected theme to the app. It adjusts colors for various components such as the header, buttons, and text based on the dark or light mode preference.

- `generate_color_palette(dark_mode, theme_file)` Generates a color palette using the primary and secondary colors from the theme, allowing for consistent styling across the app. It calculates gradient colors to create a smooth visual experience.

Key features:

- **Customizable themes**: Easily switch between light and dark modes to match user preferences.
- **Color palette generation**: Dynamically generate color gradients based on the theme, enhancing the app's visual design.
- **Seamless integration**: The theme settings are seamlessly integrated into the app, providing a consistent and customizable user experience.

This module ensures that the Streamlit application has a visually consistent theme, with customizable options for light and dark modes, contributing to an improved overall user experience.

Future Improvements

Planned updates include:

- Integration with other booking platforms (e.g., Airbnb).
- Advanced machine learning models for recommendations.
- Localization support for non-English languages.