SCRAPELY

Project Overview

BY: BRENDAN AIRD

PART 1 - Project Description

PROJECT DEFINITION

Develop an application that can be used by individuals looking to automatically scrape internet reviews from websites. This application would look at a website like TripAdvisor for a particular hotel and grab all the individual reviews posted from all pages. These collected reviews would then need to be presented with appropriate visualizations of the important data.

Based on recommendation and research: Python (various libraries), HTML, and Javascript with Flask

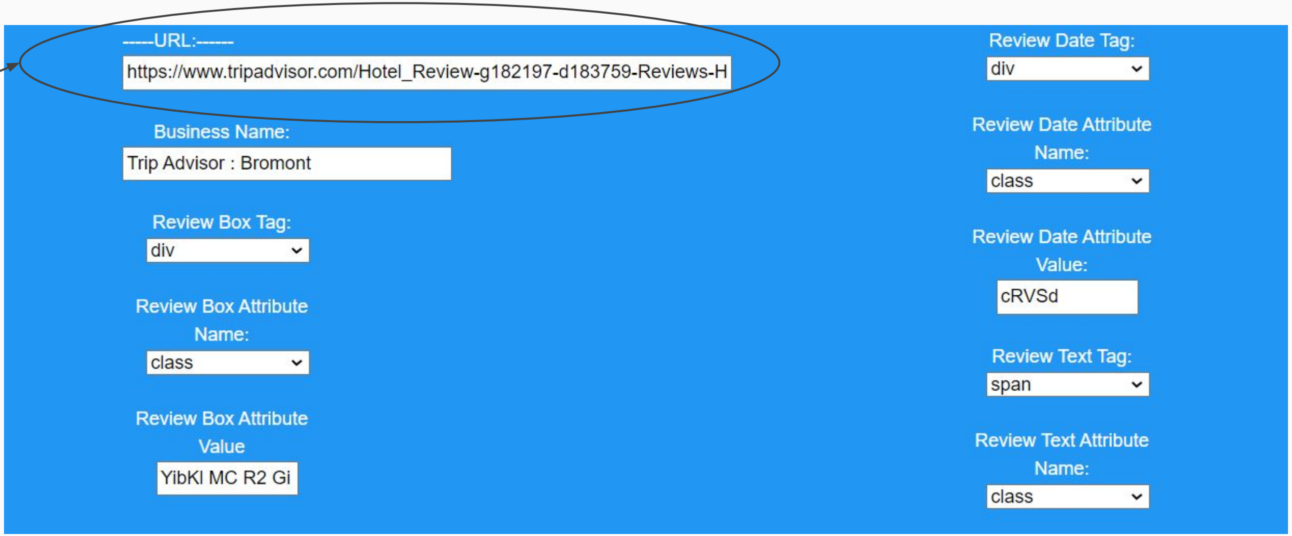
PROJECT REQUIREMENTS

1. **Research different methods to obtain review text from websites.**
2. **Allow the user to select the website/topic to scrape.**
3. Prepare the data for presentation.
4. Use visual attributes (color, size, etc.) to describe the characteristics of the data.
5. Allow the user to adjust how the data is displayed.
6. Provide options for various types of text analysis

SOLUTIONS for Requirement 1 & 2

1. **GREAT HELP! - StackOverFlow, W3Schools, Web Scraping Youtube Channels, Github, Sentry, Pydata.org, & developer.mozilla.org.**

2. **Solution to RQ: 2**



-----URL:------

https://www.tripadvisor.com/Hotel_Review-g182197-d183759-Reviews-H

Business Name: Trip Advisor : Bromont

Review Box Tag: div

Review Box Attribute Name: class

Review Box Attribute Value: YibKI MC R2 Gi

Review Date Tag: div

Review Date Attribute Name: class

Review Date Attribute Value: cRVSd

Review Text Tag: span

Review Text Attribute Name: class

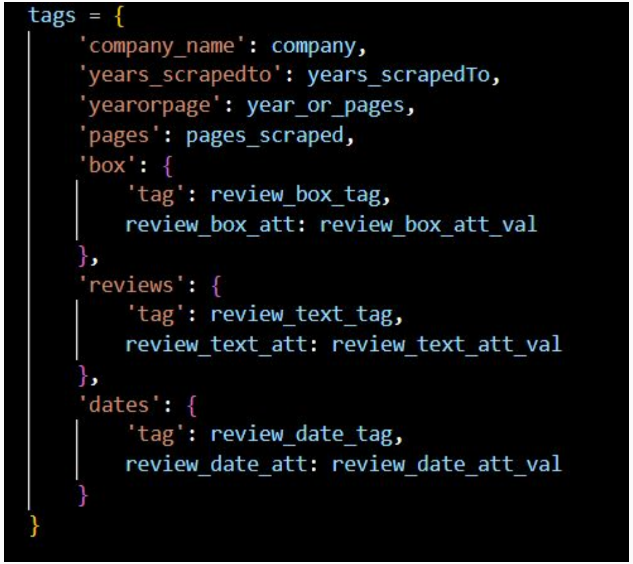
PROJECT REQUIREMENTS

1. Research different methods to obtain review text from websites.
2. Allow the user to select the website/topic to scrape.
3. **Prepare the data for presentation.**
4. **Use visual attributes (color, size, etc.) to describe the characteristics of the data.**
5. Allow the user to adjust how the data is displayed.
6. Provide options for various types of text analysis

SOLUTIONS for Requirement 3 & 4

3. Sentiment_Processing() with pandas & textblob

4. Visualizedata() with Matplotlib & seaborn

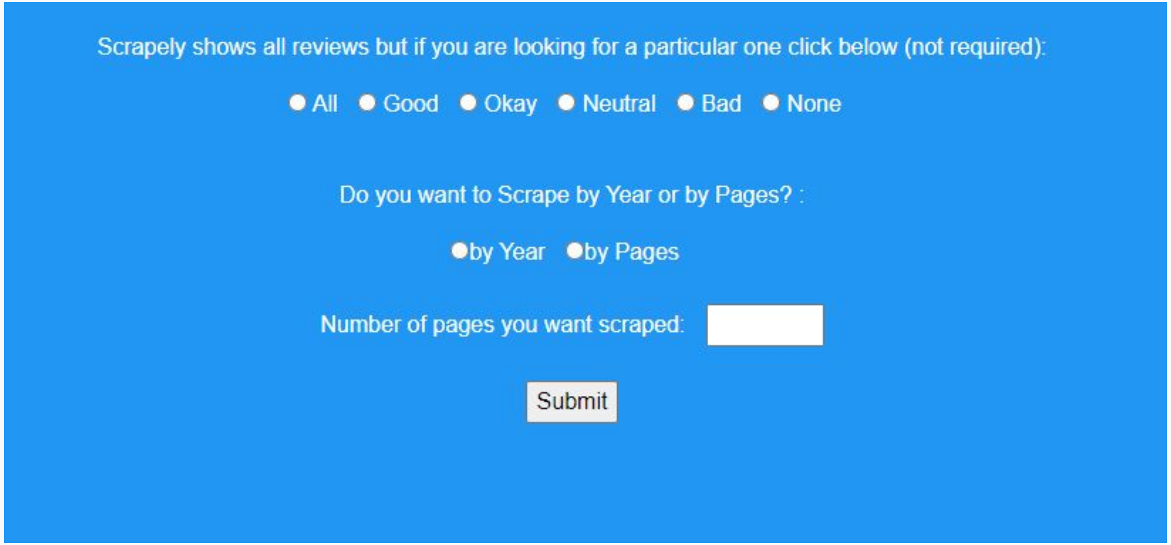


```
tags = {
    'company_name': company,
    'years_scrapedto': years_scrapedTo,
    'yearorpage': year_or_pages,
    'pages': pages_scraped,
    'box': {
        'tag': review_box_tag,
        review_box_att: review_box_att_val
    },
    'reviews': {
        'tag': review_text_tag,
        review_text_att: review_text_att_val
    },
    'dates': {
        'tag': review_date_tag,
        review_date_att: review_date_att_val
    }
}
```

PROJECT REQUIREMENTS

1. Research different methods to obtain review text from websites.
2. Allow the user to select the website/topic to scrape.
3. Prepare the data for presentation.
4. Use visual attributes (color, size, etc.) to describe the characteristics of the data.
5. **Allow the user to adjust how the data is displayed.**
6. **Provide options for various types of text analysis**

Solution to Requirement 5



Scrapely shows all reviews but if you are looking for a particular one click below (not required):

○ All ○ Good ○ Okay ○ Neutral ○ Bad ○ None

Do you want to Scrape by Year or by Pages? :

○by Year ○by Pages

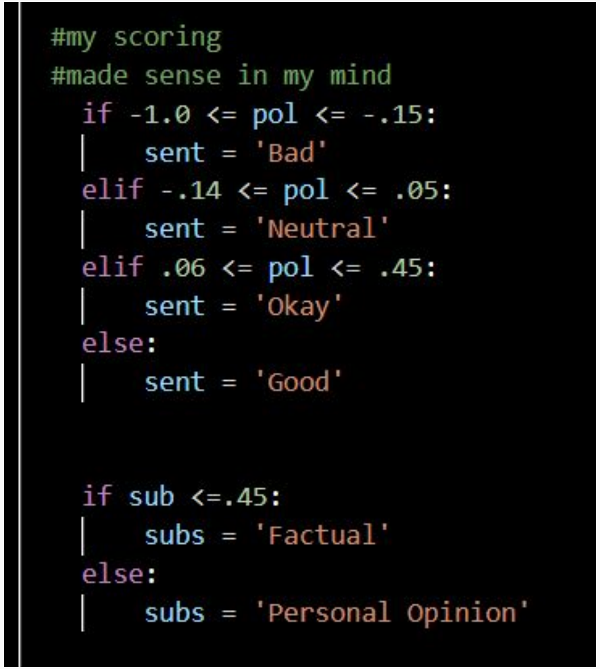
Number of pages you want scraped:

Various options for ways to adjust how the scraped data can be displayed!

Solution to Requirement 6: Sentiment Analysis

My Scoring:

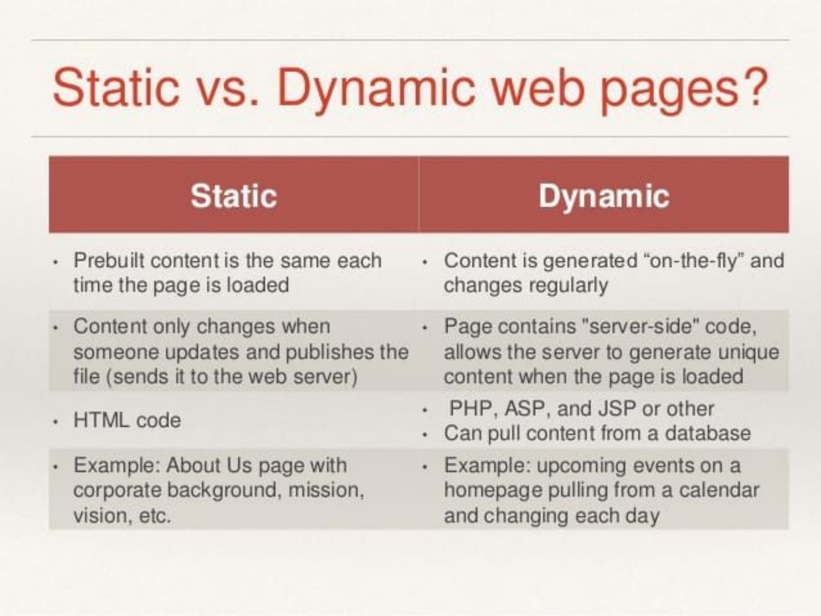
- Polarity - Positive, Neutral, Negative
  - Positive = “I **love** this hotel”
  - Negative = “I **hate** this hotel”
  - Neutral = “My stay was **fine** at this hotel”
- Subjectivity - Personal Opinion, or Factual
  - Personal Opinion - “**I think** this hotel is great”
  - Factual - “**This hotel is located in De Pere**”



```
#my scoring
#made sense in my mind
  if -1.0 <= pol <= -.15:
      sent = 'Bad'
  elif -.14 <= pol <= .05:
      sent = 'Neutral'
  elif .06 <= pol <= .45:
      sent = 'Okay'
  else:
      sent = 'Good'


  if sub <=.45:
      subs = 'Factual'
  else:
      subs = 'Personal Opinion'
```

Exceptions



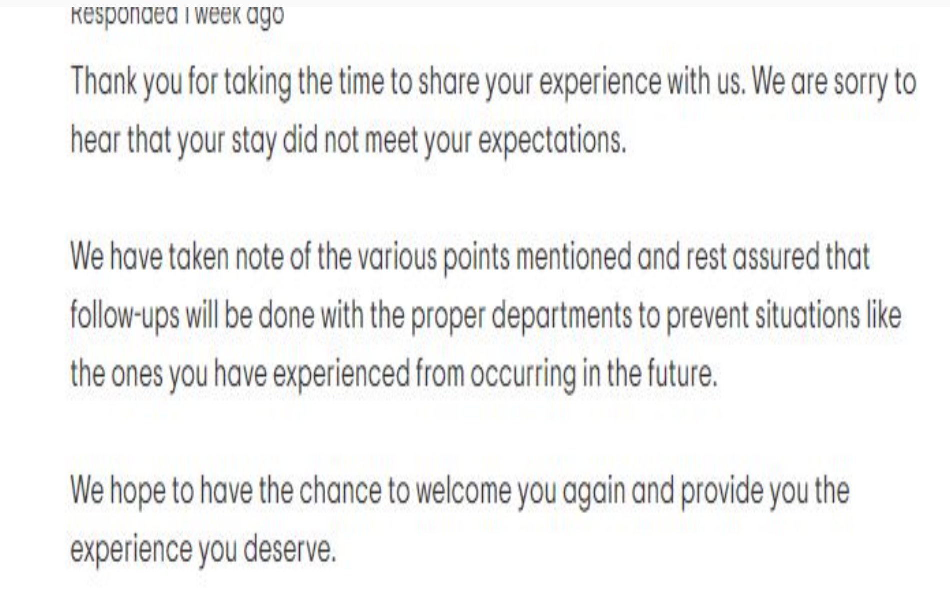
Static vs. Dynamic web pages?

Static	Dynamic
• Prebuilt content is the same each time the page is loaded	• Content is generated “on-the-fly” and changes regularly
• Content only changes when someone updates and publishes the file (sends it to the web server)	• Page contains "server-side" code, allows the server to generate unique content when the page is loaded
• HTML code	• PHP, ASP, and JSP or other • Can pull content from a database
• Example: About Us page with corporate background, mission, vision, etc.	• Example: upcoming events on a homepage pulling from a calendar and changing each day

1. Only can scrape Static websites (BeautifulSoup)

- Problem: Javascript

My Methods



Responded 1 week ago

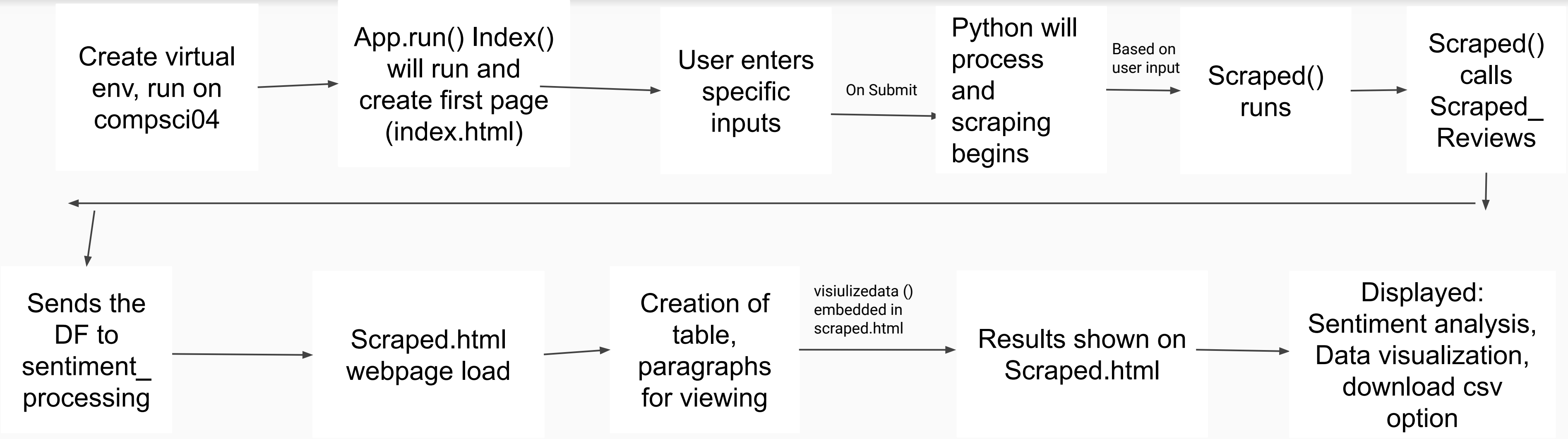
Thank you for taking the time to share your experience with us. We are sorry to hear that your stay did not meet your expectations.

We have taken note of the various points mentioned and rest assured that follow-ups will be done with the proper departments to prevent situations like the ones you have experienced from occurring in the future.

We hope to have the chance to welcome you again and provide you the experience you deserve.

1. HTML PARSING PROCESS - BeautifulSoup
2. Design (UI): set up specific inputs for users to interact with (containers)
3. Experiments: used my scraping tool on dynamic websites (Yelp), or website that have inconsistent html (EBay)

FLOW AND DESIGN FOR MY PROJECT



Create virtual env, run on compsci04

App.run() Index() will run and create first page (index.html)

User enters specific inputs

On Submit

Python will process and scraping begins

Based on user input

Scraped() runs

Scraped() calls Scraped_ Reviews

Sends the DF to sentiment_ processing

Scraped.html webpage load

Creation of table, paragraphs for viewing

visiulizedata () embedded in scraped.html

Results shown on Scraped.html

Displayed: Sentiment analysis, Data visualization, download csv option

Part 2 -Live Demonstration - PREPARE ON FRONT END

Part 3 - Learning and Development Process

Strategies & Resources

How I found most of my answers I was looking for:

1. stackoverflow / W3schools
2. A LOT OF RESEARCH THROUGH GOOGLE
3. Youtube Scraping Channels
4. Various discussion with Dr. Diederich (Flask, Sentiment Analysis, UI Design)

Extensions for Future Students

1. Using Selenium with puppeteer instead of BeautifulSoup
2. Progress Bar
3. Having my web application formatted right on a mobile device
4. Using a proxy server so the users IP Address does not get banned.

PART 4 - Q & A (BE HAPPY TO ANSWER THEM!)

BACK UP IF SOMETHING GOES WRONG