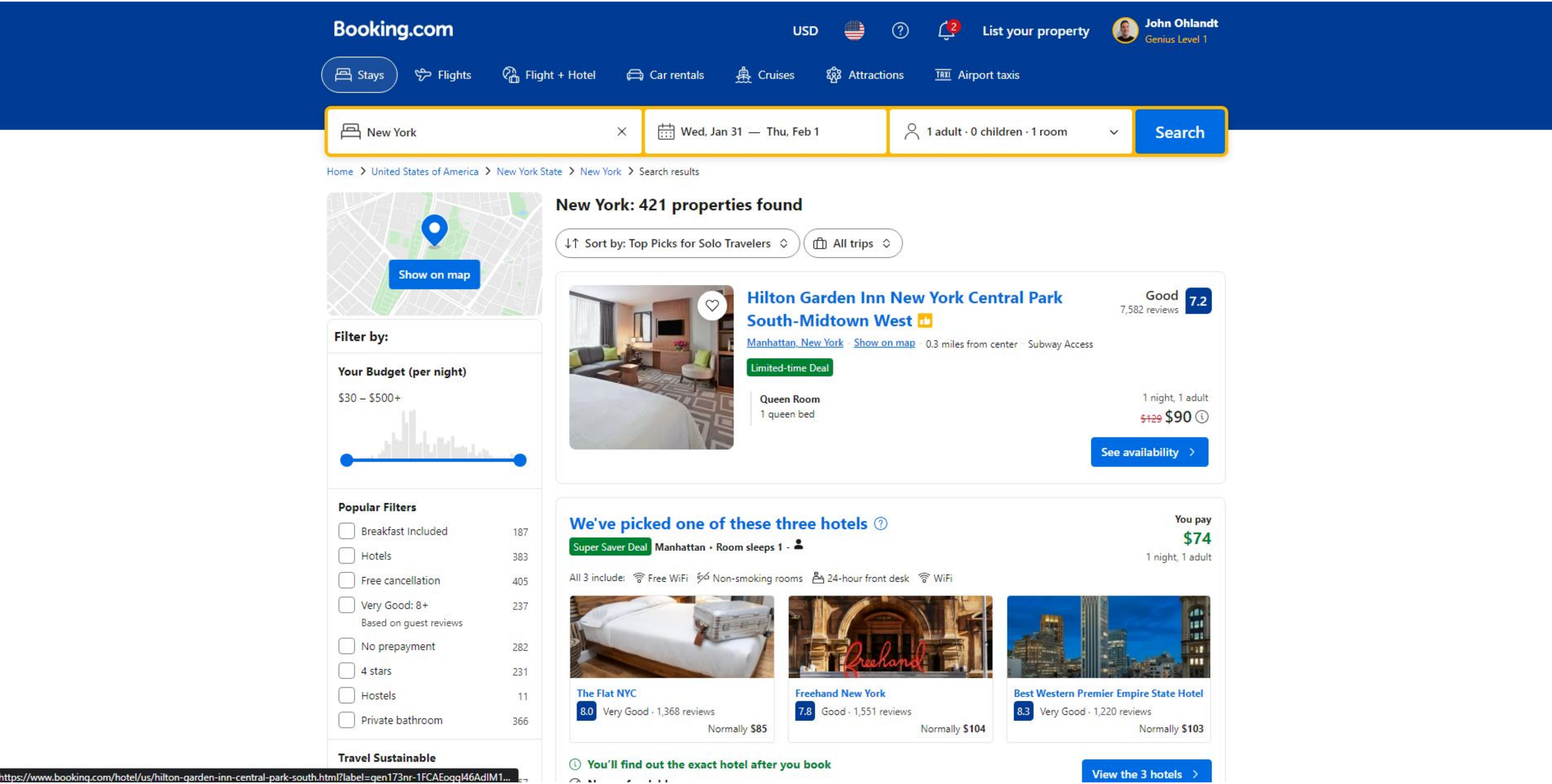
Data Scraping Booking.com

By JT Ohlandt

Use Cases

- Trip Planning
  - Whether or not a hotel is a good deal
- Competitor Research
  - Competing companies could utilize data in analysis
- Market Research
  - Analyzing prices or other data across a time period



Booking.com

USD

List your property

John Ohlandt
Genius Level 1

Stays · Flights · Flight + Hotel · Car rentals · Cruises · Attractions · Airport taxis

New York ✕

Wed, Jan 31 — Thu, Feb 1

1 adult · 0 children · 1 room

Search

Home > United States of America > New York State > New York > Search results

Show on map

Filter by:

Your Budget (per night)

$30 – $500+

Popular Filters

□ Breakfast Included	187
□ Hotels	383
□ Free cancellation	405
□ Very Good: 8+ Based on guest reviews	237
□ No prepayment	282
□ 4 stars	231
□ Hostels	11
□ Private bathroom	366

Travel Sustainable

New York: 421 properties found

Sort by: Top Picks for Solo Travelers

All trips

Hilton Garden Inn New York Central Park South-Midtown West

Good 7.2
7,582 reviews

Manhattan, New York · Show on map · 0.3 miles from center · Subway Access

Limited-time Deal

Queen Room
1 queen bed

1 night, 1 adult
~~$129~~ $90

See availability >

We've picked one of these three hotels

Super Saver Deal Manhattan • Room sleeps 1 -

You pay
$74
1 night, 1 adult

All 3 include: Free WiFi · Non-smoking rooms · 24-hour front desk · WiFi

The Flat NYC	Freehand New York	Best Western Premier Empire State Hotel
8.0 Very Good · 1,368 reviews	7.8 Good · 1,551 reviews	8.3 Very Good · 1,220 reviews
Normally $85	Normally $104	Normally $103

You'll find out the exact hotel after you book

View the 3 hotels >

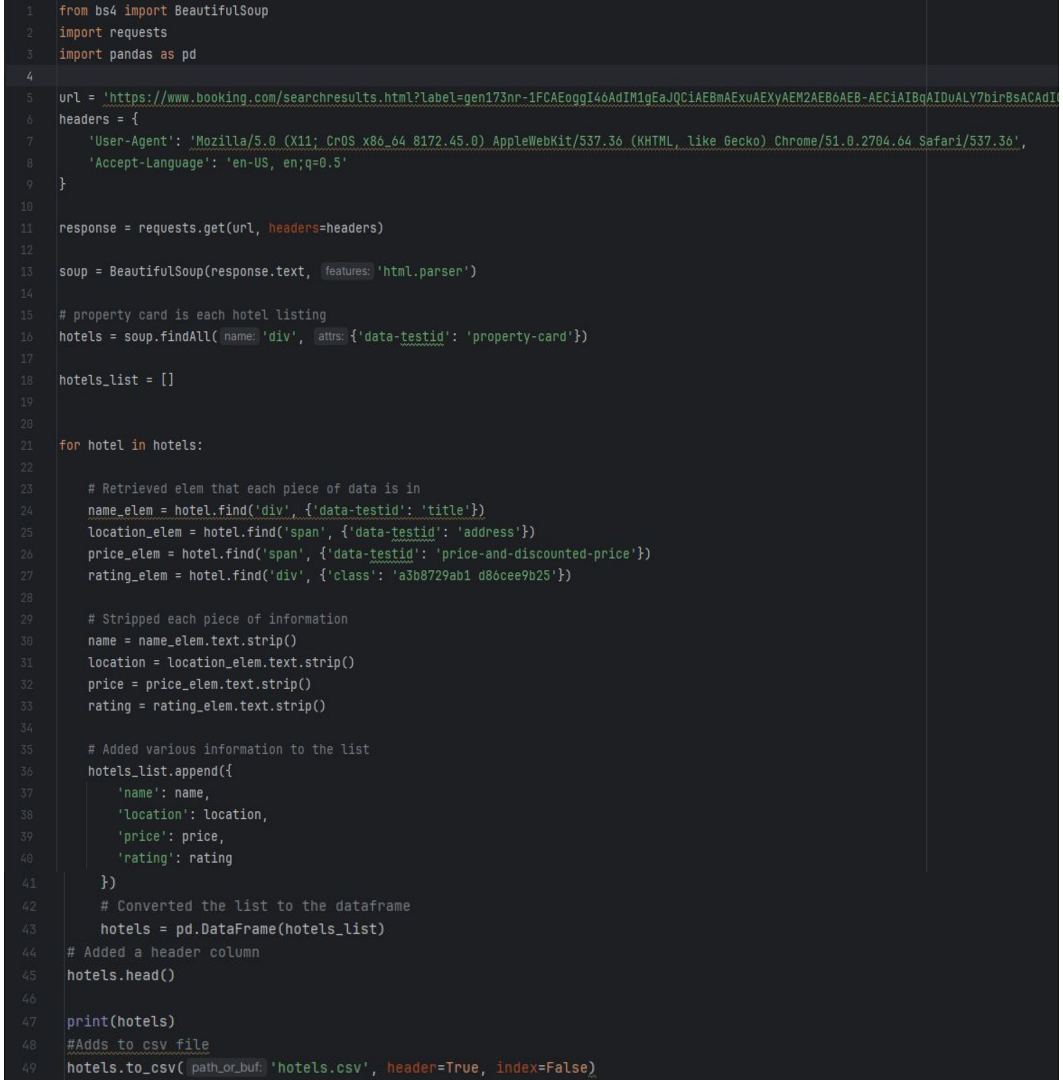
https://www.booking.com/hotel/us/hilton-garden-inn-central-park-south.html?label=gen173nr-1FCAEoggI46AdIM1...

Software

- Beautiful Soup
    - Used to Parse through HTML
- Pandas
    - For dataframe and modifying data
- Chromedriver & Selenium
    - Simulate web page and access more pages

Data Scraper

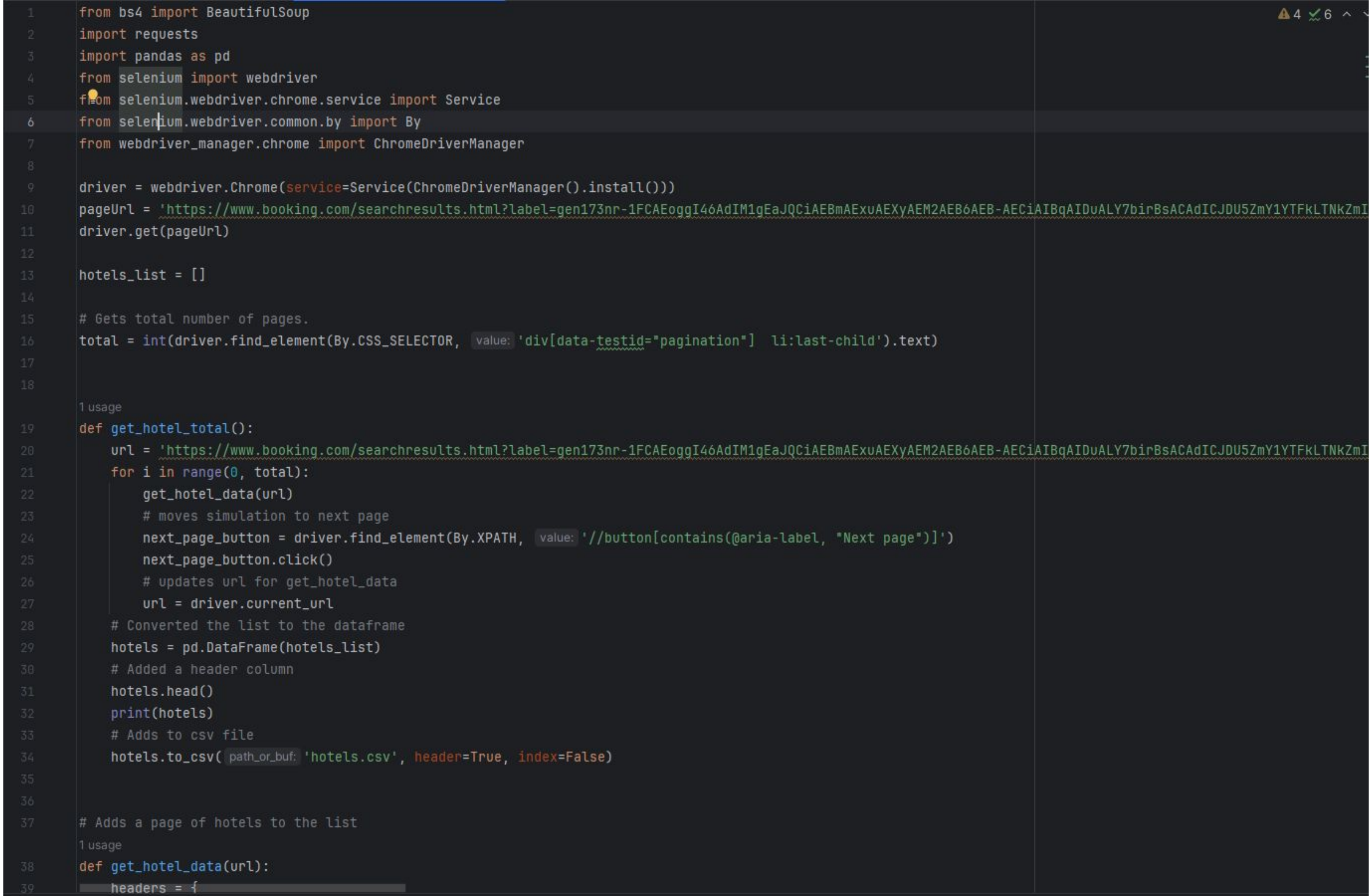
- Went through each hotel listing using find
- Recorded data with strip function
- Added them to a list
- Used panda to store them in a dataframe
- Finally saved as CSV



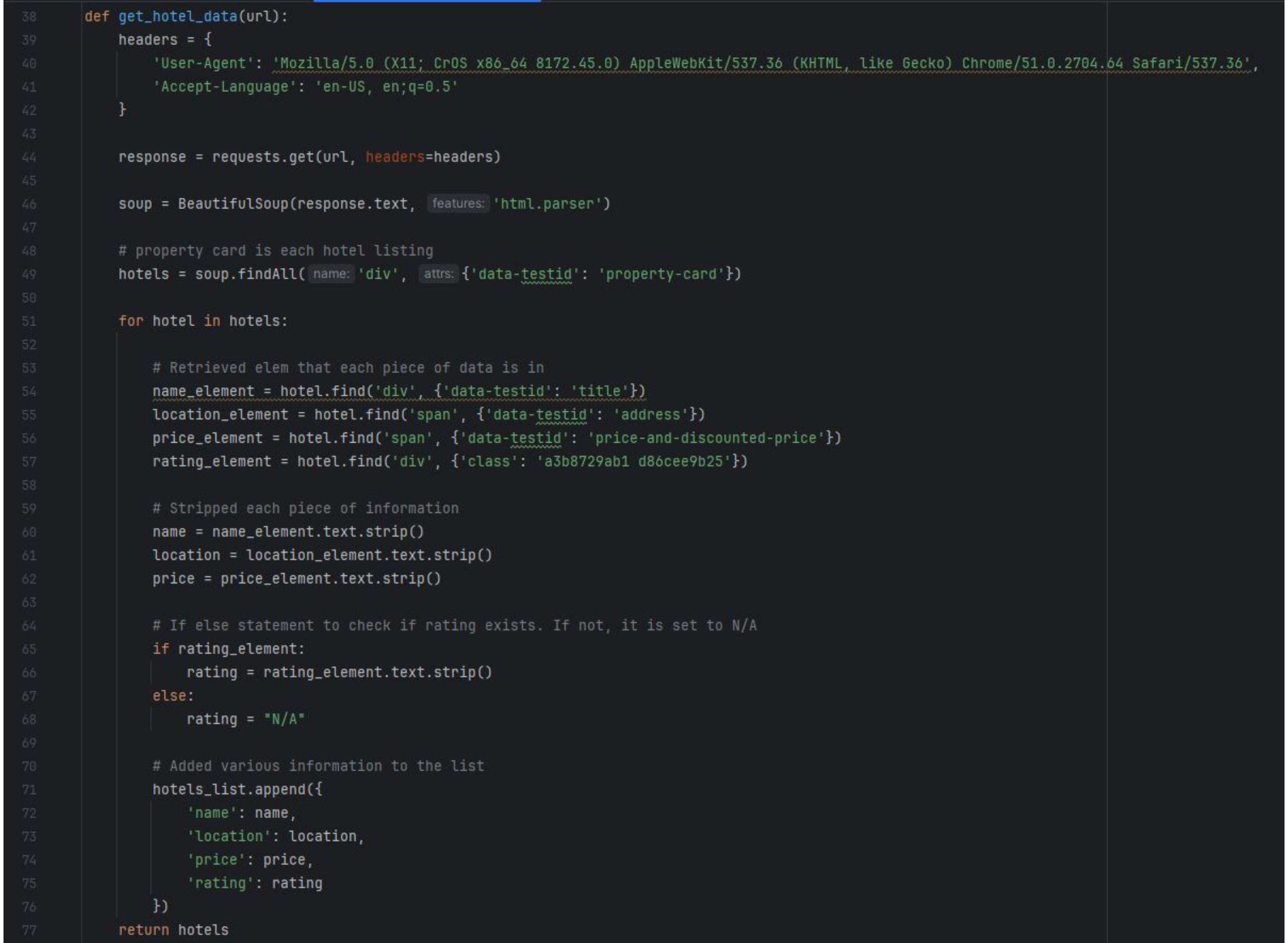
```python
from bs4 import BeautifulSoup
import requests
import pandas as pd

url = 'https://www.booking.com/searchresults.html?label=gen173nr-1FCAEoggI46AdIM1gEaJQCiAEBmAExuAEXyAEM2AEB6AEB-AECiAIBqAIDuALY7birBsACAdI
headers = {
    'User-Agent': 'Mozilla/5.0 (X11; CrOS x86_64 8172.45.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.64 Safari/537.36',
    'Accept-Language': 'en-US, en;q=0.5'
}

response = requests.get(url, headers=headers)

soup = BeautifulSoup(response.text, features: 'html.parser')

# property card is each hotel listing
hotels = soup.findAll( name: 'div',  attrs: {'data-testid': 'property-card'})

hotels_list = []


for hotel in hotels:

    # Retrieved elem that each piece of data is in
    name_elem = hotel.find('div', {'data-testid': 'title'})
    location_elem = hotel.find('span', {'data-testid': 'address'})
    price_elem = hotel.find('span', {'data-testid': 'price-and-discounted-price'})
    rating_elem = hotel.find('div', {'class': 'a3b8729ab1 d86cee9b25'})

    # Stripped each piece of information
    name = name_elem.text.strip()
    location = location_elem.text.strip()
    price = price_elem.text.strip()
    rating = rating_elem.text.strip()

    # Added various information to the list
    hotels_list.append({
        'name': name,
        'location': location,
        'price': price,
        'rating': rating
    })
    # Converted the list to the dataframe
    hotels = pd.DataFrame(hotels_list)
# Added a header column
hotels.head()

print(hotels)
#Adds to csv file
hotels.to_csv( path_or_buf: 'hotels.csv', header=True, index=False)
```

Handling Additional Pages and Handling No Reviews

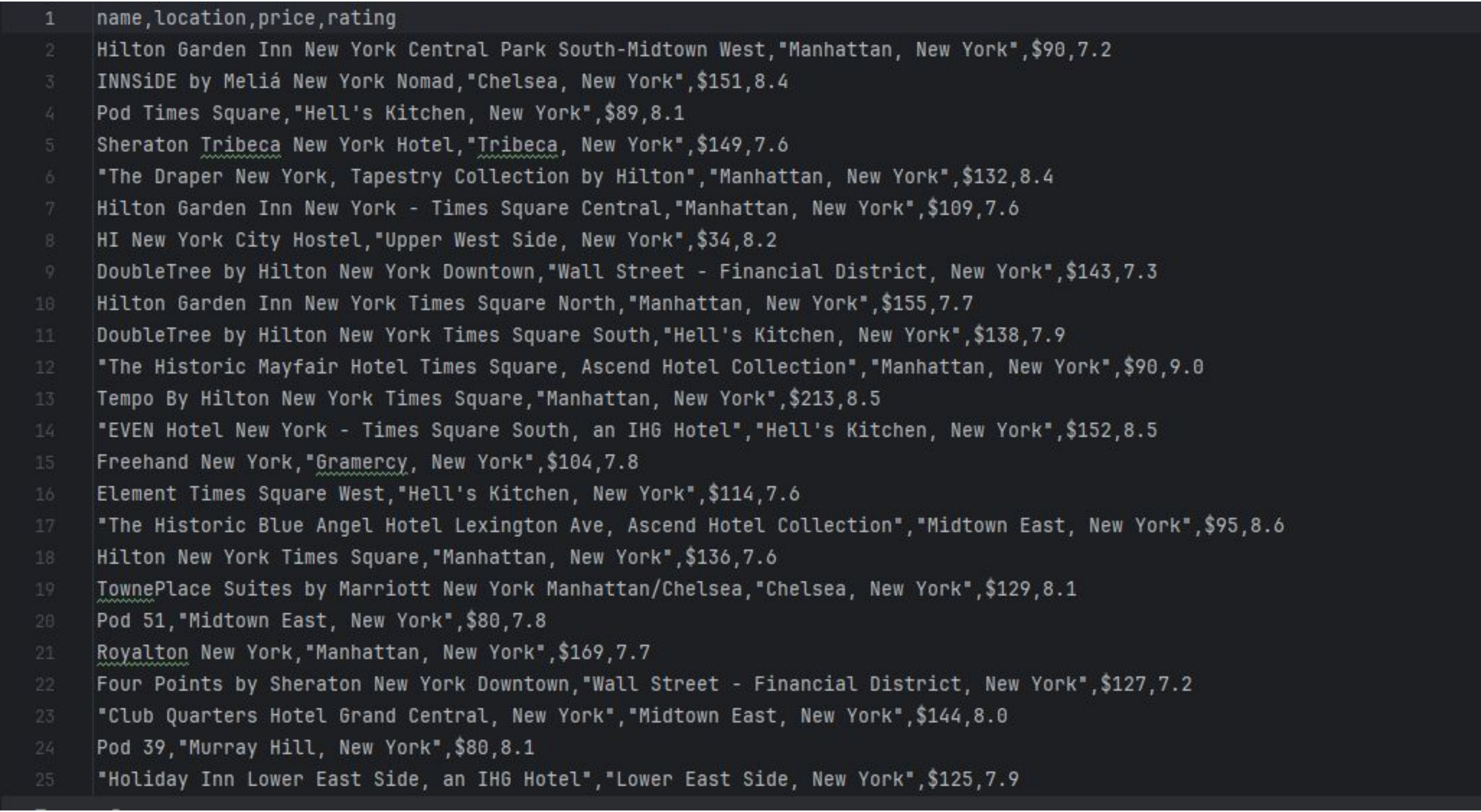
- Broke into web scraper into 2 different functions
- Utilized Selenium and ChromeDriver to go through each page
- Putting N/A for no reviews



```python
from bs4 import BeautifulSoup
import requests
import pandas as pd
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
pageUrl = 'https://www.booking.com/searchresults.html?label=gen173nr-1FCAEoggI46AdIM1gEaJQCiAEBmAExuAEXyAEM2AEB6AEB-AECiAIBqAIDuALY7birBsACAdICJDU5ZmY1YTFkLTNkZmI
driver.get(pageUrl)

hotels_list = []

# Gets total number of pages.
total = int(driver.find_element(By.CSS_SELECTOR, 'div[data-testid="pagination"]  li:last-child').text)


def get_hotel_total():
    url = 'https://www.booking.com/searchresults.html?label=gen173nr-1FCAEoggI46AdIM1gEaJQCiAEBmAExuAEXyAEM2AEB6AEB-AECiAIBqAIDuALY7birBsACAdICJDU5ZmY1YTFkLTNkZmI
    for i in range(0, total):
        get_hotel_data(url)
        # moves simulation to next page
        next_page_button = driver.find_element(By.XPATH, '//button[contains(@aria-label, "Next page")]')
        next_page_button.click()
        # updates url for get_hotel_data
        url = driver.current_url
    # Converted the list to the dataframe
    hotels = pd.DataFrame(hotels_list)
    # Added a header column
    hotels.head()
    print(hotels)
    # Adds to csv file
    hotels.to_csv('hotels.csv', header=True, index=False)


# Adds a page of hotels to the list
def get_hotel_data(url):
    headers = {
```



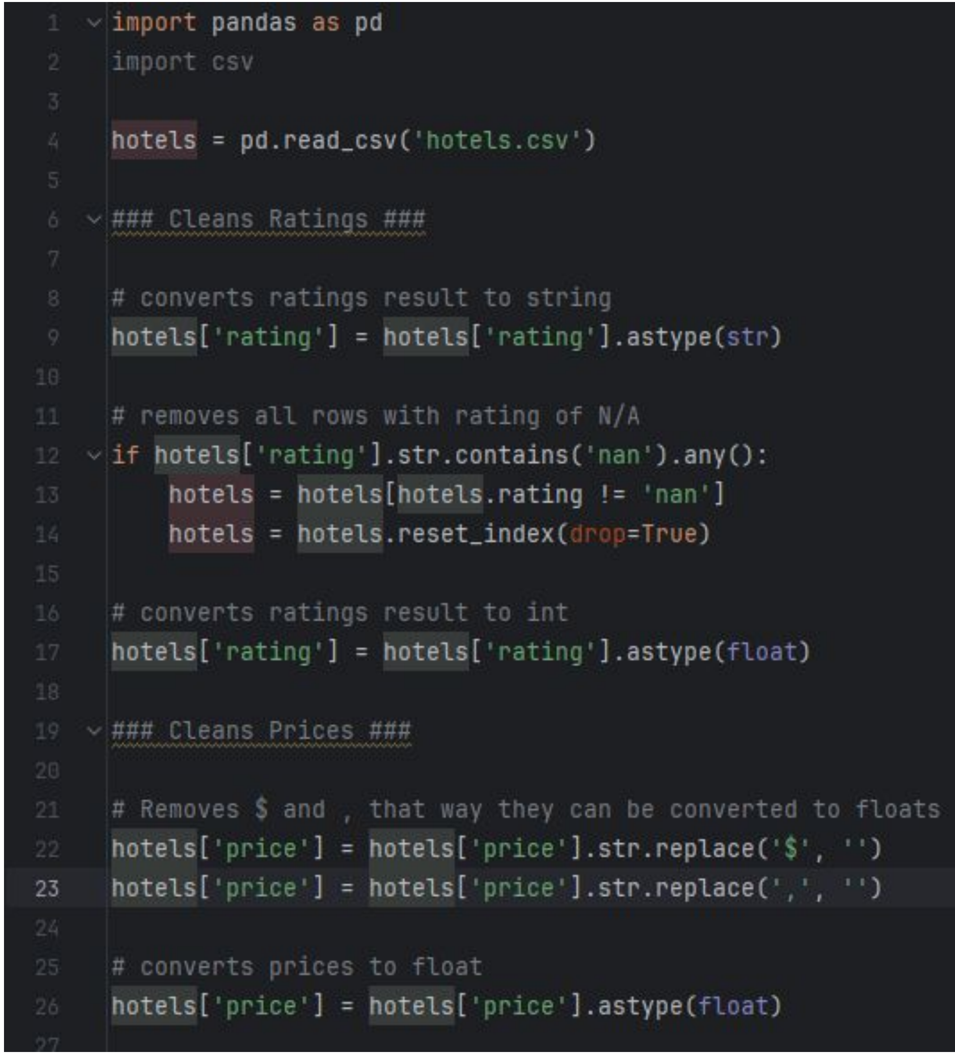
```python
def get_hotel_data(url):
    headers = {
        'User-Agent': 'Mozilla/5.0 (X11; CrOS x86_64 8172.45.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.64 Safari/537.36',
        'Accept-Language': 'en-US, en;q=0.5'
    }

    response = requests.get(url, headers=headers)

    soup = BeautifulSoup(response.text,  features: 'html.parser')

    # property card is each hotel listing
    hotels = soup.findAll( name: 'div',  attrs: {'data-testid': 'property-card'})

    for hotel in hotels:

        # Retrieved elem that each piece of data is in
        name_element = hotel.find('div', {'data-testid': 'title'})
        location_element = hotel.find('span', {'data-testid': 'address'})
        price_element = hotel.find('span', {'data-testid': 'price-and-discounted-price'})
        rating_element = hotel.find('div', {'class': 'a3b8729ab1 d86cee9b25'})

        # Stripped each piece of information
        name = name_element.text.strip()
        location = location_element.text.strip()
        price = price_element.text.strip()

        # If else statement to check if rating exists. If not, it is set to N/A
        if rating_element:
            rating = rating_element.text.strip()
        else:
            rating = "N/A"

        # Added various information to the list
        hotels_list.append({
            'name': name,
            'location': location,
            'price': price,
            'rating': rating
        })
    return hotels
```



```
name,location,price,rating
Hilton Garden Inn New York Central Park South-Midtown West,"Manhattan, New York",$90,7.2
INNSiDE by Meliá New York Nomad,"Chelsea, New York",$151,8.4
Pod Times Square,"Hell's Kitchen, New York",$89,8.1
Sheraton Tribeca New York Hotel,"Tribeca, New York",$149,7.6
"The Draper New York, Tapestry Collection by Hilton","Manhattan, New York",$132,8.4
Hilton Garden Inn New York - Times Square Central,"Manhattan, New York",$109,7.6
HI New York City Hostel,"Upper West Side, New York",$34,8.2
DoubleTree by Hilton New York Downtown,"Wall Street - Financial District, New York",$143,7.3
Hilton Garden Inn New York Times Square North,"Manhattan, New York",$155,7.7
DoubleTree by Hilton New York Times Square South,"Hell's Kitchen, New York",$138,7.9
"The Historic Mayfair Hotel Times Square, Ascend Hotel Collection","Manhattan, New York",$90,9.0
Tempo By Hilton New York Times Square,"Manhattan, New York",$213,8.5
"EVEN Hotel New York - Times Square South, an IHG Hotel","Hell's Kitchen, New York",$152,8.5
Freehand New York,"Gramercy, New York",$104,7.8
Element Times Square West,"Hell's Kitchen, New York",$114,7.6
"The Historic Blue Angel Hotel Lexington Ave, Ascend Hotel Collection","Midtown East, New York",$95,8.6
Hilton New York Times Square,"Manhattan, New York",$136,7.6
TownePlace Suites by Marriott New York Manhattan/Chelsea,"Chelsea, New York",$129,8.1
Pod 51,"Midtown East, New York",$80,7.8
Royalton New York,"Manhattan, New York",$169,7.7
Four Points by Sheraton New York Downtown,"Wall Street - Financial District, New York",$127,7.2
"Club Quarters Hotel Grand Central, New York","Midtown East, New York",$144,8.0
Pod 39,"Murray Hill, New York",$80,8.1
"Holiday Inn Lower East Side, an IHG Hotel","Lower East Side, New York",$125,7.9
```

Cleaning Up Data

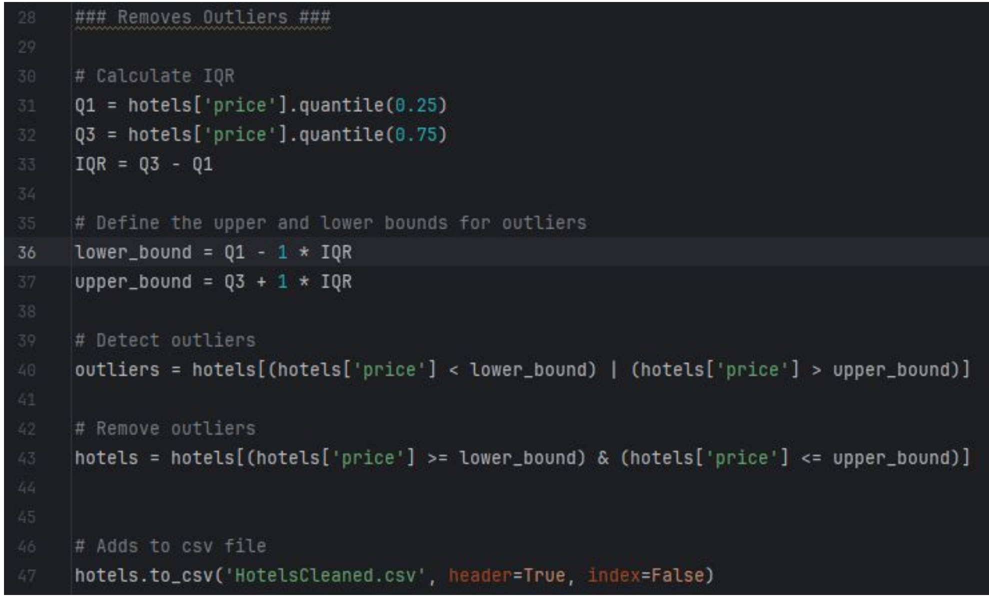
- Converted Ratings and Price to Floats
- Removed N/A Ratings



```python
import pandas as pd
import csv

hotels = pd.read_csv('hotels.csv')

### Cleans Ratings ###

# converts ratings result to string
hotels['rating'] = hotels['rating'].astype(str)

# removes all rows with rating of N/A
if hotels['rating'].str.contains('nan').any():
    hotels = hotels[hotels.rating != 'nan']
    hotels = hotels.reset_index(drop=True)

# converts ratings result to int
hotels['rating'] = hotels['rating'].astype(float)

### Cleans Prices ###

# Removes $ and , that way they can be converted to floats
hotels['price'] = hotels['price'].str.replace('$', '')
hotels['price'] = hotels['price'].str.replace(',', '')

# converts prices to float
hotels['price'] = hotels['price'].astype(float)
```

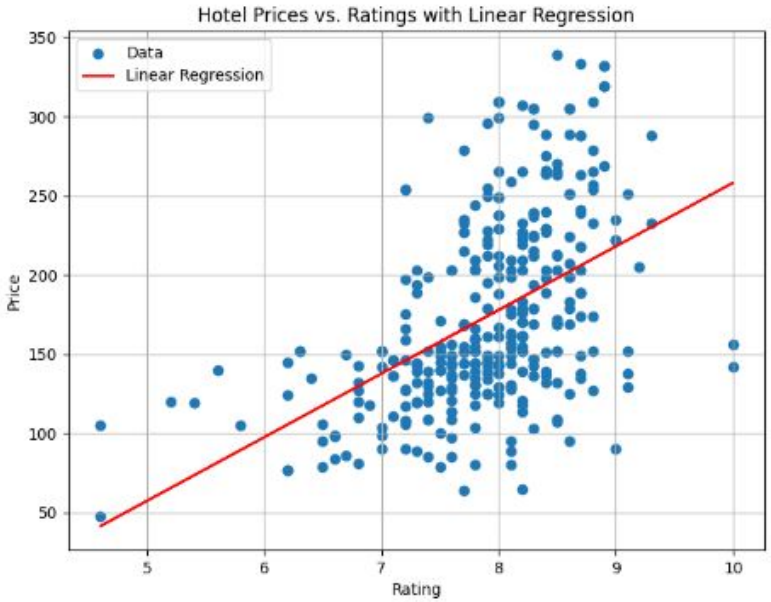
Cleaning Data

- Removed Outliers
- Created New CSV File

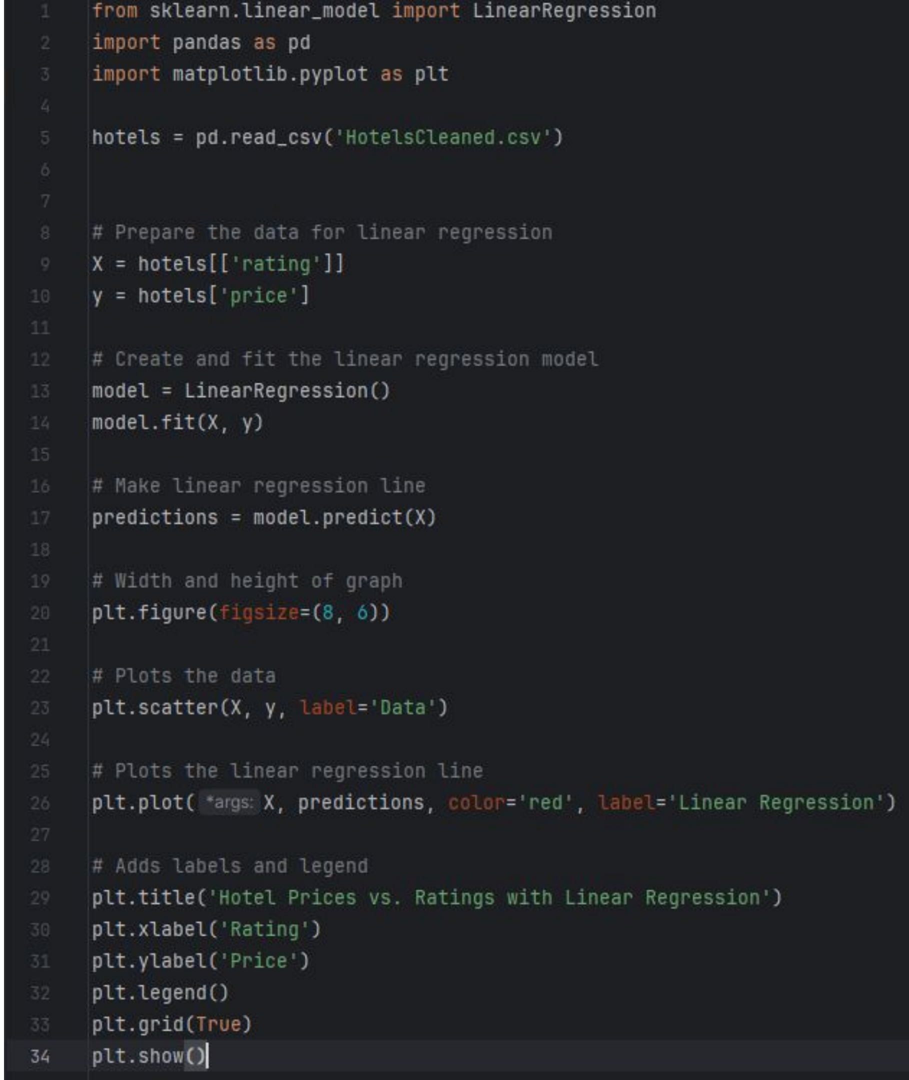


```
### Removes Outliers ###

# Calculate IQR
Q1 = hotels['price'].quantile(0.25)
Q3 = hotels['price'].quantile(0.75)
IQR = Q3 - Q1

# Define the upper and lower bounds for outliers
lower_bound = Q1 - 1 * IQR
upper_bound = Q3 + 1 * IQR

# Detect outliers
outliers = hotels[(hotels['price'] < lower_bound) | (hotels['price'] > upper_bound)]

# Remove outliers
hotels = hotels[(hotels['price'] >= lower_bound) & (hotels['price'] <= upper_bound)]


# Adds to csv file
hotels.to_csv('HotelsCleaned.csv', header=True, index=False)
```

Linear Regression Model



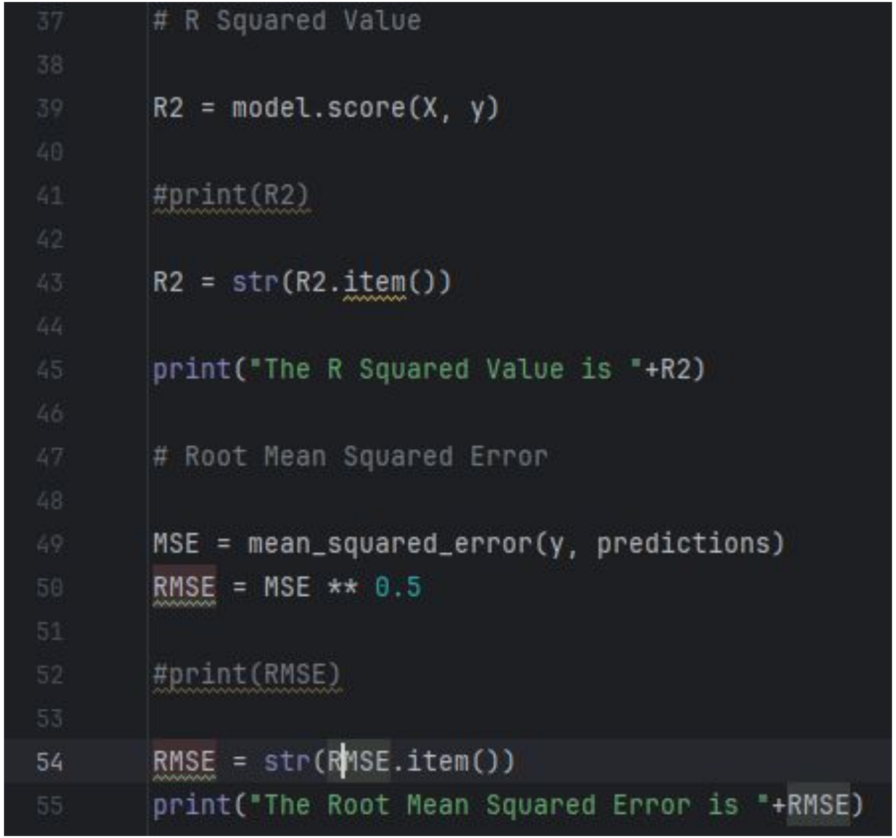
Hotel Prices vs. Ratings with Linear Regression



```
from sklearn.linear_model import LinearRegression
import pandas as pd
import matplotlib.pyplot as plt

hotels = pd.read_csv('HotelsCleaned.csv')


# Prepare the data for linear regression
X = hotels[['rating']]
y = hotels['price']

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X, y)

# Make linear regression line
predictions = model.predict(X)

# Width and height of graph
plt.figure(figsize=(8, 6))

# Plots the data
plt.scatter(X, y, label='Data')

# Plots the linear regression line
plt.plot(X, predictions, color='red', label='Linear Regression')

# Adds labels and legend
plt.title('Hotel Prices vs. Ratings with Linear Regression')
plt.xlabel('Rating')
plt.ylabel('Price')
plt.legend()
plt.grid(True)
plt.show()
```

R Squared and Root Mean Squared

- R squared value of 0.234
  - Model could be improved
- RMSE value of $52



```
# R Squared Value

R2 = model.score(X, y)

#print(R2)

R2 = str(R2.item())

print("The R Squared Value is "+R2)

# Root Mean Squared Error

MSE = mean_squared_error(y, predictions)
RMSE = MSE ** 0.5

#print(RMSE)

RMSE = str(RMSE.item())
print("The Root Mean Squared Error is "+RMSE)
```