Project Proposal: Real Estate Analysis with Web Scraping and Pandas

Project members: Y. Sachith Nimesh

Description:

Real estate is a key indicator of economic growth and can be used to make informed investment decisions. In this project, we will collect data on real estate prices and trends in a specific area using web scraping and analyze it to identify patterns and predict future trends. We will use Python, the Pandas library, and web scraping tools like BeautifulSoup to scrape data from websites like Zillow, Redfin, or Realtor.com.

The project will consist of the following steps:

Define the specific area and type of real estate to focus on, such as single-family homes, apartments, or commercial properties.

Scrape data on real estate prices, square footage, number of bedrooms and bathrooms, and other relevant features using web scraping tools like BeautifulSoup.

Clean and preprocess the data using Pandas, removing any duplicates or invalid entries, and transforming it into a format suitable for analysis.

Perform exploratory data analysis to identify patterns and trends in the data, such as correlations between price and square footage or location.

Use statistical and machine learning models to predict future trends in real estate prices based on historical data, taking into account factors such as economic indicators and demographic changes.

Visualize the results using tools like Matplotlib or Seaborn to create clear and informative plots and graphs.

The final output of the project will be a report summarizing the key findings and insights, including visualizations of the data and statistical analyses. This report can be used by investors or real estate professionals to make informed decisions about buying or selling properties in the target area.

In [2]:

```
# Import libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

In [17]:

```
# Define the URL and headers
url = "https://www.zillow.com/homes/for_sale/New-York-NY_rb/"
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 S

# Send a GET request to the URL and retrieve the HTML content
response = requests.get(url, headers=headers)
content = response.content
```

In [16]:

```
# Parse the HTML content with BeautifulSoup
soup = BeautifulSoup(content, "html.parser")

# Find all the listings on the page
listings = soup.find_all("div", class_="list-card-info")

# Create a list to store the data
data = []

# Loop through each listing and extract the relevant information
 for listing in listings:
 address = listing.find("address").get_text().strip()
 price = listing.find("div", class_="list-card-price").get_text().strip()
 details = listing.find_all("ul", class_="list-card-details")[0].find_all("li")
 beds = details[0].get_text().strip().split()[0]
 baths = details[1].get_text().strip().split()[0]
 sqft = details[2].get_text().strip().split()[0]
 data.append([address, price, beds, baths, sqft])

data
```

In [10]:

```
# Convert the list to a Pandas DataFrame
df = pd.DataFrame(data, columns=["Address", "Price", "Beds", "Baths", "Sqft"])

# Clean up the data
df["Price"] = df["Price"].str.replace("$", "").str.replace(",", "").astype(int)
df["Beds"] = df["Beds"].str.replace(" bd", "").astype(int)
df["Baths"] = df["Baths"].str.replace(" ba", "").astype(float)
df["Sqft"] = df["Sqft"].str.replace(",", "").str.replace(" sqft", "").astype(int)

# Print the DataFrame
print(df)
```