Dokumentacja Projektu

Web Scraping z wykorzystaniem Beautiful Soup oraz Django

Projekt Webowe Biblioteki Programistyczne semestr VI

Wojciech Kozieł

Politechnika Śląska

1. Założenia projektu

Projekt zakłada napisanie aplikacji mającej na celu pobieranie danych ze strony otodom.pl i przygotowanie statystyk na ich podstawie. Do utworzenia projektu zostanie wykorzystany program Visual Studio Code, a cała aplikacja zostanie utworzona z wykorzystaniem biblioteki **Django** języka Python oraz biblioteki **Beautiful** Soup, mającej na celu pobieranie danych ze strony.

2. Opis projektu

Projekt składa się z jednej aplikacji o nazwie /app, zawiera w sobie jeden widok zawierający wszystkie tworzone statystyki. Dane ze strony otodom.pl pobierane są za pomocą biblioteki Beautiful Soup i zapisywane do zmiennej.

```
site = requests.get("https://www.otodom.pl/wynajem/mieszkanie/?nrAdsPerPage=72")
soup = BeautifulSoup(site.content, 'html.parser')
```

Następnie ze zmiennej soup pobierane ją poszczególne dane takie jak:

- tytuł oferty
- miasto
- dzielnica miasta
- cena
- metraż mieszkaniu
- ilość pokoi w mieszkaniu

Funkcja pobierająca konkretne dane ze zmiennej soup - przykład:

```
def getOfferTitle():
    titles = [i.get_text() for i in
soup.select('span.offer-item-title')]
    return titles
```

Następnie dane zapisywane są w obiekcie DataFrame biblioteki **pandas:**

```
def getData():
    data = pd.DataFrame()
    places = getPlace()
    data['Tytuł'] = getOfferTitle()
    data['Miasto'] = getCities(places)
    data['Dzielnica'] = getDistricts(places)
    data['Cena'] = getPrices()
    data['Metraż'] = getYardage()
    data['Pokoje'] = getRooms()
```

Tak spreparowane dane są gotowe aby można było z nich utworzyć wykresy.

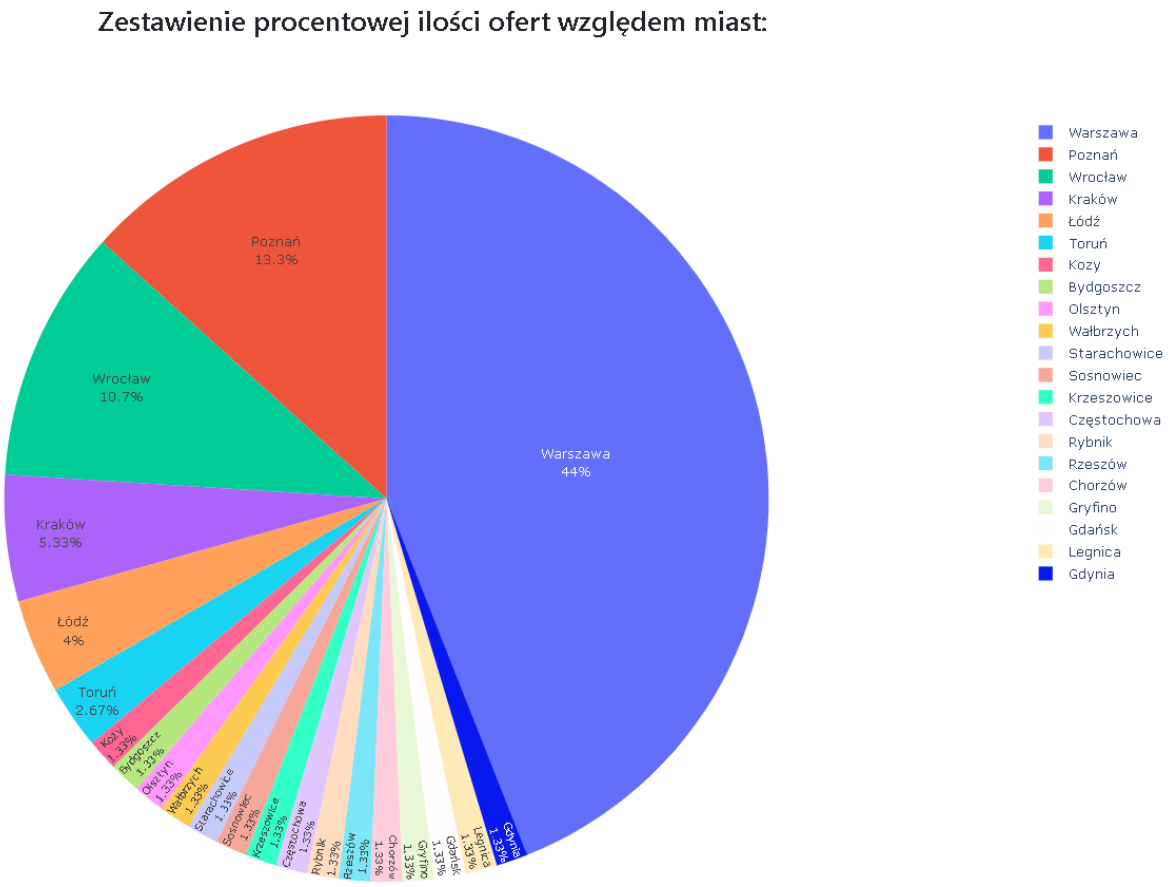
Tabela 1. Tabela zawierająca dane statystyczne uzyskane na podstawie danych ze scrapowania

Dane	#
Ilość zescrapowanych ofert	75
Najczęściej występujące miasto	Warszawa
Najczęściej występująca dzielnica	Warszawa, Śródmieście
Najtańsza oferta	898.0 zł/mc
Najdroższa oferta	12000.0 zł/mc
Średnia cena mieszkania	2266.64 zł/mc
Średnia cena za m^2	47.34 zł
Średni metraż mieszkania	47.88 m^2
Średnia ilość pokoi	2.0

Dane do tabeli uzyskane zostały za pomocą funkcji dostępnych dla obiektu DataFrame biblioteki pandas. Funkcje te pozwalają na:

- obliczenie ile razy obiekt pojawia się w kolumnie - count()
- znalezienie najczęsciej występującej wartości - value_counts() oraz idxmax()
- znalezienie największej i najmniejszej wartości - max() oraz min()
- obliczenie średniej - mean()

Wykres 1. Zestawienie ilości ofert w danych miastach za pomocą wykresu kołowego:

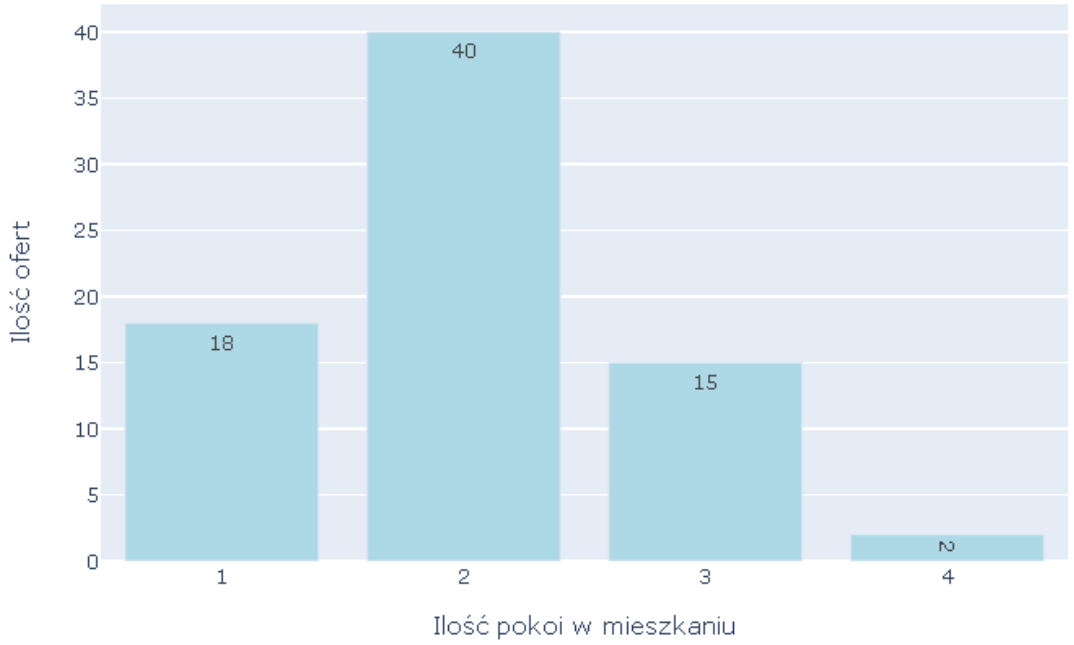


Do utworzenia zestawu danych dla wykresu wykorzystana została funkcja **value_counts()** biblioteki pandas, która z wybranej kolumny listę zawierającą ilość wystąpień danej pozycji w kolumnie co w tym przypadku jest ilością wystąpień każdego z miast w zescrapowanych danych.

Kod źródłowy wykresu wygląda następująco:

```
def drawCityDiagram():
    df = pd.DataFrame(data['Miasto'].value_counts())
    df.columns = ['Ilosc']
    df.index.name = 'Miasto'
    fig = px.pie(df, values="Ilosc", names=df.index, height=800)
    fig.update_traces(textposition='inside',
textinfo='percent+label')
    plot_div = plot(fig, output_type='div')
    return plot_div
```

Wykres 2. Zestawia ilość ofert dla każdej z ilości dostępnych pokoi

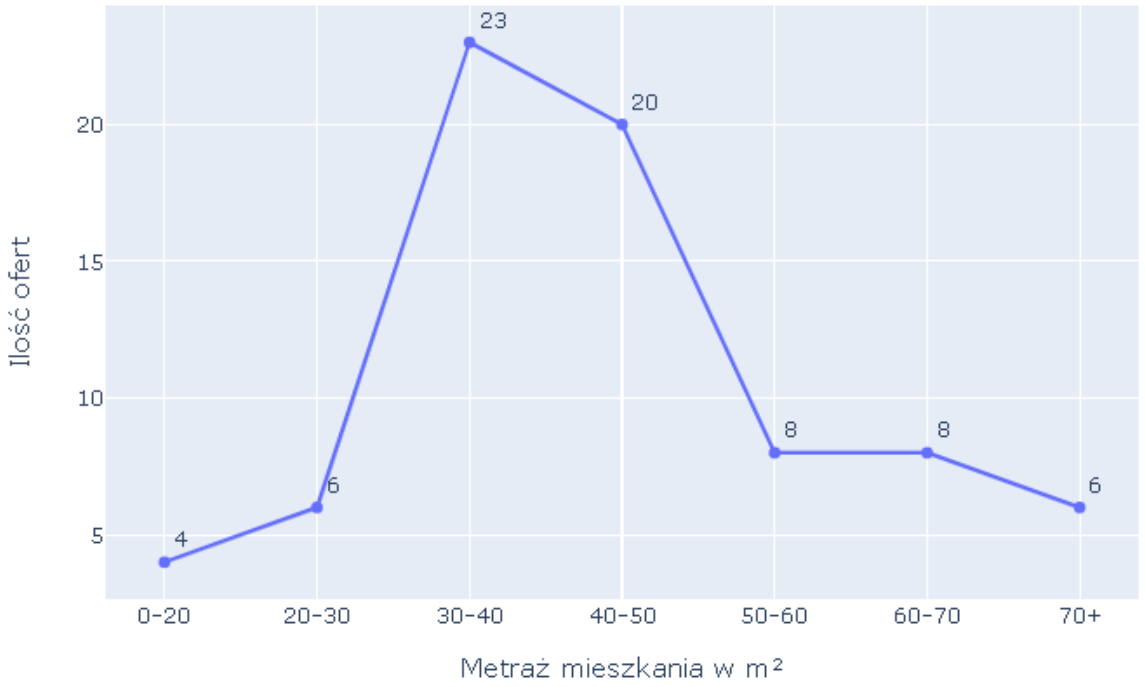


Statystykę dla wykresu uzyskano poprzez wykorzystanie funkcji value_counts() dostępnej w ramach biblioteki pandas.

Kod:

```
def drawRoomDiagram():
    df = pd.DataFrame(data['Pokoje'].value_counts())
    df.columns = ['Ilość ofert']
    df.index.name = "Ilość pokoi w mieszkaniu"
    fig = px.bar(df, x=df.index, y='Ilość ofert', text='Ilość ofert')
    fig.update_traces(marker_color='lightblue')
    fig.update_layout(showlegend=False)
    plot_div = plot(fig, output_type='div')
    return plot_div
```

Wykres 3. Zestawienie średniego metrażu do ilości ofert

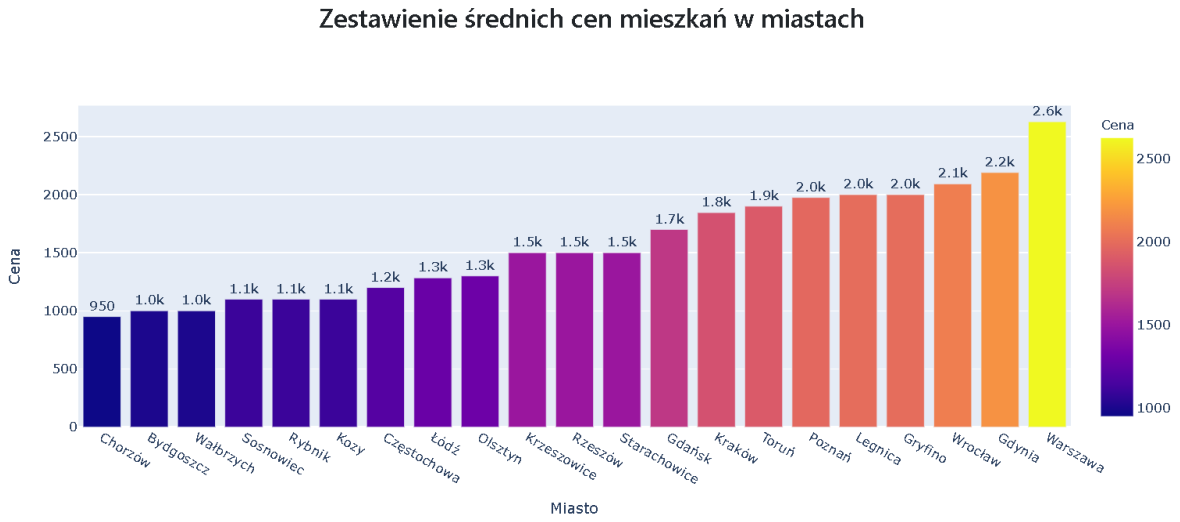


Dane do wykresu uzyskano poprzez utworzenie pętli sprawdzającej poszczególne przedziały metrażu i zapisującej odpowiednie ilości do słownika, który następnie za pomocą funkcji from_dict() biblioteki pandas został zamieniony w DataFrame i wykorzystany do utworzenia powyższego wykresu.

Kod:

```
def drawYardageDiagram():
    D = getYards()
    df = pd.DataFrame.from_dict(D, orient='index')
    df.index.name = 'Metraż mieszkania w m²'
    df.columns=['Ilość ofert']
    fig = px.line(df, x=df.index, y='Ilość ofert', text='Ilość ofert')
    fig.update_traces(textposition='top right')
    fig.update_layout(showlegend=False)
    plot_div = plot(fig, output_type='div')
    return plot_div
```

Wykres 4.



Powyższy wykres uzyskano z pomocą funkcji grupowania danych oraz funkcji mean() biblioteki pandas. Dane zostały posortowane od najmniejszej do największej i wyświetlone z wykorzystaniem parametru color względem wartości dostępnej w ramach biblioteki plotly express.

Kod:

```
def drawPriceDiagram():
    df = data.groupby('Miasto').mean().reset_index()
    df = df.sort_values(by=['Cena'])
    fig = px.bar(df, x='Miasto', y='Cena', text='Cena', color='Cena')
    fig.update_traces(texttemplate='%{text:.2s}',
textposition='outside')
    fig.update_layout(showlegend=False,)
    plot_div = plot(fig, output_type='div')
    return plot_div
```