

Studium Magisterskie

Kierunek: Analiza Danych – Big Data

Bartosz Radź

Nr albumu 116066

**Pozyskanie oraz wizualizacja danych warszawskiego rynku nieruchomości z wykorzystaniem technik ETL i Power BI**

Praca magisterska

napisana w InstytucieEkonometrii

pod kierunkiem naukowym

dr Sebastiana Zająca

Warszawa 2023.

Spis treści

[Wstęp 5](#_Toc136470098)

[Rozdział I. Proces ETL dla pobrania danych oraz ich wizualizacja 7](#_Toc136470099)

[I.1 Pojęcie ETL 7](#_Toc136470100)

[I.2 Przykłady zastosowań procesu ETL 10](#_Toc136470101)

[I.3 Web scraping jako sposób pozyskiwania danych 12](#_Toc136470102)

[I.4 Wizualizacja danych jako narzędzie do analiz informacji 17](#_Toc136470103)

[Rozdział II. Analiza serwisu otodom.pl jako źródło danych 24](#_Toc136470104)

[II.1 Charakterystyka serwisu 24](#_Toc136470105)

[II.2 Funkcjonalności oraz dane 25](#_Toc136470106)

[Rozdział III. Projekt procesu ETL 32](#_Toc136470107)

[III.1 Koncepcja i założenia 32](#_Toc136470108)

[III.2 Narzędzia i technologie 33](#_Toc136470109)

[III.2.1.1 Python 33](#_Toc136470110)

[III.2.1.2 Visual Code Studio (IDE) 34](#_Toc136470111)

[III.2.1.3 Selenium 35](#_Toc136470112)

[III.2.1.4 SQLite 36](#_Toc136470113)

[III.2.1.5 Apache Airflow 37](#_Toc136470114)

[III.2.1.6 Power BI 38](#_Toc136470115)

[Rozdział IV. Implementacja 41](#_Toc136470116)

[IV.1 Pozyskanie danych 41](#_Toc136470117)

[IV.2 Transformacja danych 46](#_Toc136470118)

[IV.3 Ładowanie danych 48](#_Toc136470119)

[IV.4 Automatyzacja procesu 50](#_Toc136470120)

[IV.5 Wizualizacja danych 52](#_Toc136470121)

[Zakończenie 55](#_Toc136470122)

[LITERATURA 57](#_Toc136470123)

[Spis wykresów 59](#_Toc136470124)

[Spis rysunków 59](#_Toc136470125)

[Streszczenie 61](#_Toc136470126)

[Summary 61](#_Toc136470127)

Wstęp

W dzisiejszych czasach ogromne ilości danych są generowane, gromadzone   
i udostępniane w różnych źródłach online. Dane te mogą być niezwykle wartościowe dla różnych branż i dziedzin, w tym również dla rynku nieruchomości. W przypadku analizy rynku mieszkaniowego, dostęp do aktualnych i dokładnych danych jest kluczowy dla podejmowania trafnych decyzji biznesowych oraz lepszego zrozumienia trendów i dynamiki rynku.

Na podstawie badań Narodowego Banku Polskiego można stwierdzić, że a rynku pierwotnym w Warszawie można zaobserwować utrzymującą się tendencję wzrostową cen mieszkań za metr kwadratowy powierzchni użytkowej. W pierwszym kwartale 2023 roku liczba ofert mieszkań wyniosła około 12,3 tysiąca, co stanowi niewielki spadek w porównaniu do czwartego kwartału 2022 roku. W tym okresie sprzedano około 2,5 tysiąca mieszkań. Średnia cena ofertowa mieszkań w stolicy wyniosła 13302 zł/mkw, a cena transakcyjna wyniosła 12861 zł/mkw. Najwyższe ceny za metr kwadratowy obserwowano w dzielnicach Ochota i Wola, natomiast najniższe ceny występowały w Rembertowie, Wawrze i Wesołej. Mieszkania o powierzchni do 40 mkw osiągały najwyższe średnie ceny zarówno ofertowe, jak i transakcyjne, mimo to sprzedawały się one najszybciej. Największą liczbę mieszkań zgłoszonych do sprzedaży stanowiły mieszkania o powierzchni 40-60 mkw, które były także najchętniej kupowane. Średni czas sprzedaży mieszkań deweloperskich wyniósł 11,8 miesiąca i wydłużył się o 1,4 miesiąca w porównaniu z poprzednim okresem[[1]](#footnote-1).

Na rynku wtórnym w Warszawie również utrzymuje się tendencja wzrostowa cen za metr kwadratowy mieszkania. Ponad 45% ofert stanowiły mieszkania z ceną powyżej 13,5 tysiąca złotych za metr kwadratowy. Najwyżej wyceniane mieszkania pod względem metrażu znajdowały się w Śródmieściu, na Żoliborzu, Wilanowie i Mokotowie. Najpopularniejsze lokalizacje dla nabywców to Ochota, Wola i Mokotów. Podobnie jak na rynku pierwotnym, najwięcej ofert i transakcji dotyczyło mieszkań o powierzchni 40-60 mkw. Czas sprzedaży mieszkań na rynku wtórnym nieznacznie skrócił się do 3,8 miesiąca[[2]](#footnote-2).

W obecnej erze cyfrowej, gdy ludzie mają możliwość szukania mieszkań na sprzedaż za pośrednictwem Internetu, platformy sprzedaży nieruchomości zdobyły ogromną popularność zarówno wśród kupujących, jak i sprzedających. Są one niezwykle przydatnym narzędziem umożliwiającym łatwe i wygodne przeglądanie ofert. Ponadto są one również źródłem danych, które przy odpowiednich technikach programistycznych można pozyskiwać i wykorzystać do analizy rynku.

Web scraping, czyli pozyskiwanie danych z witryn internetowych, jest techniką, która umożliwia automatyczne pobieranie danych z różnych stron internetowych. W niniejszej pracy, technika ta zostanie wykorzystana do pozyskiwania informacji dotyczących mieszkań na sprzedaż w Warszawie. Pobieranie danych zrealizowane zostało z serwisu otodom.pl. Serwis ten stanowi jedno z najpopularniejszych źródeł informacji o nieruchomościach w Polsce, zawierając bogate i zróżnicowane dane na temat ofert sprzedaży mieszkań.

Wizualizacja danych jest narzędziem, które umożliwia łatwiejsze zrozumienie   
i analizę zgromadzonych informacji. W ramach pracy magisterskiej zostanie opracowany interaktywny dashboard, który pozwoli na przejrzystą wizualizację danych dotyczących mieszkań na sprzedaż w Warszawie. Dashboard ten będzie zawierał wykresy, mapy i filtry, umożliwiając użytkownikom łatwe eksplorowanie danych, analizę trendów oraz wyciąganie wniosków.

Głównym celem pracy jest dostarczenie wartościowego narzędzia dla osób zainteresowanych rynkiem mieszkań w stolicy Polski, poprzez stworzenie kompletnego procesu ETL (Extract, Transform, Load) dla danych ze strony otodom.pl. Proces ten obejmuje wydobycie danych z witryny, przekształcenie ich do odpowiedniego formatu oraz załadowanie ich do docelowej bazy danych a następnie ich wizualizację.

W dalszej części pracy przedstawione zostaną etapy procesu ETL, implementacja web scrapingu, sposób przekształcania i ładowania danych, a także projekt i implementacja interaktywnego dashboardu. Ponadto, omówione zostaną wyzwania i trudności związane   
z pozyskiwaniem i przetwarzaniem danych z otodom.pl oraz przedstawione zostaną propozycje dalszych usprawnień i rozszerzeń projektu.

# Proces ETL dla pobrania danych oraz ich wizualizacja

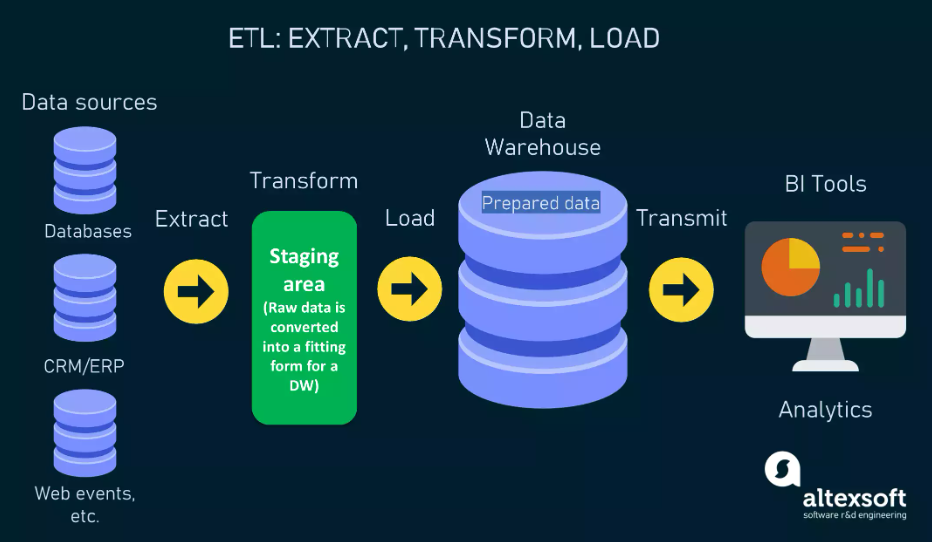
Poniższy rozdział skupia się na trzech ważnych aspektach analizy danych. Procesie ETL, webscrapingu i wizualizacji danych. W pierwszym podrozdziale omówione zostanie, czym jest proces ETL oraz jakie korzyści niesie ze sobą jego stosowanie w kontekście analizy danych. W drugim podrozdziale, zostaną przedstawione przykłady zastosowań procesu ETL   
w różnych dziedzinach biznesowych. Kolejnym ważnym aspektem, który zostanie omówiony w tej części, jest webscraping jako sposób pozyskania danych. Przedstawione zostaną narzędzia i techniki, które umożliwiają automatyczne pobieranie danych z różnych źródeł internetowych. Ostatni podrozdział skupi się na roli wizualizacji w procesie analizy danych. Omówione zostaną techniki i narzędzia do wizualizacji, które pozwalają na skuteczną prezentację informacji i zrozumienie złożonych zbiorów danych.

## Pojęcie ETL

Skrót ETL (ang. Extraction, Transformation, Loading) oznacza wyodrębnianie, przekształcanie i ładowanie. Jest to proces integracji danych, który łączy dane   
z wielu źródeł danych w jeden, spójny magazyn danych, który jest ładowany do np. hurtowni danych lub innego systemu docelowego, co przestawia poniższa grafika. Wraz z rosnącą popularnością baz danych w latach 70-tych, proces ETL został wprowadzony jako sposób na integrację i ładowanie danych do obliczeń i analiz, stając się ostatecznie główną metodą przetwarzania danych dla projektów hurtowni danych. ETL stanowi podstawę dla procesów analitycznych i uczenia maszynowego. Poprzez serię reguł biznesowych, proces ten oczyszcza i organizuje dane w sposób, który odpowiada określonym potrzebom dotyczącym BI (ang. business intelligence), takim jak miesięczne raportowanie. Może także zajmować się bardziej zaawansowaną analizą, dzięki czemu może poprawić procesy backendowe lub doświadczenia użytkowników końcowych. ETL jest często wykorzystywany przez organizacje w celu[[3]](#footnote-3):

* Pobieranie danych z dotychczasowych systemów,
* Oczyszczanie danych w celu poprawy ich jakości i ustalenia spójności,
* Załadowanie danych do bazy danych.

Rysunek 1. Przedstawienie procesu ETL.

**Źródło:** ETL overview <https://www.altexsoft.com/blog/etl-vs-elt/> [19-03-2023].

Podczas wyodrębniania dane surowe są kopiowane lub eksportowane z lokalizacji źródłowych do obszaru etapowego (np. tymczasowa baza danych lub pamięć podręczna komputera). Zespoły zarządzające danymi mogą pobierać je z różnych źródeł, które mogą zawierać ustrukturowane lub nieustrukturowane dane. Źródła te mogą obejmować np[[4]](#footnote-4):

* serwery SQL lub NoSQL,
* systemu CRM lub ERP,
* wiadomości email,
* strony internetowe,
* api (ang. application programming interface),
* pliki takie jak np.: csv, xslx, txt.

Podczas przekształcania danych, kiedy są one w obszarze etapowym, następuje ich transformacja oraz konsolidowanie z myślą o ich zamierzonej analizie. Ta faza może obejmować wiele działań takich jak[[5]](#footnote-5):

* Filtrowanie, oczyszczanie, usuwanie duplikatów, weryfikacja i uwierzytelnianie danych.
* Wykonywanie obliczeń, tłumaczeń lub podsumowań na podstawie surowych danych. Może to obejmować zmianę nagłówków wierszy i kolumn dla spójności, konwersję walut lub innych jednostek miary, edycję ciągów tekstowych i wiele innych.
* Przeprowadzanie audytów w celu zapewnienia jakości danych i zgodności.
* Usuwanie, szyfrowanie lub ochrona danych objętych regulacjami branżowymi lub rządowymi.
* Formatowanie danych do tabel lub tabel połączonych w celu dopasowania schematu docelowej hurtowni danych.

Podczas ładowania danych, które jest ostatnim etapem ETL, przekształcone dane są przenoszone z obszaru etapowego do docelowej lokalizacji. Zazwyczaj obejmuje to początkowe wczytanie wszystkich danych, a następnie okresowe wczytywanie zmian danych inkrementalnych i rzadziej, pełne odświeżanie w celu usunięcia i zastąpienia danych   
w magazynie danych. Dla większości organizacji korzystających z ETL, proces jest zautomatyzowany, dobrze zdefiniowany, ciągły lub sterowany partiami. Zwykle ETL odbywa się w godzinach poza „szczytem”, gdy ruch na systemach źródłowych i magazynie danych jest najniższy.[[6]](#footnote-6)

Można wyróżnić wiele rodzajów ETL, jednak potrzeby biznesowe zwykle powodują, że najczęściej używanymi są[[7]](#footnote-7):

* Przetwarzanie partiami (ang. batch processing): Dane są pobierane i przetwarzane   
  w określonych partiach, zazwyczaj w określonych odstępach czasu, np. co godzinę lub co dzień.
* Przetwarzanie w czasie rzeczywistym (ang. streaming processing): Dane są przetwarzane natychmiast po ich pobraniu ze źródłowego systemu, umożliwiając szybsze i bardziej aktualne wyniki przetwarzania.

Podsumowując, proces ETL składa się z trzech etapów jakimi są pozyskanie, przekształcenie oraz importowanie danych do jednej lokalizacji. Każdy z tych etapów może być przeprowadzony w różny sposób. Podobnie do jego podprocesów, ETL można przeprowadzić na wiele sposobów w zależności od potrzeb biznesowych.

## Przykłady zastosowań procesu ETL

Proces ETL jest powszechnie stosowany w różnych obszarach biznesowych do integrowania, przekształcania i konsolidowania danych z różnych źródeł w jedną, ujednoliconą bazę danych lub hurtownię danych. Poniżej przedstawiono 5 przykładów[[8]](#footnote-8).

Pierwszą dziedziną biznesową w której powszechnym jest budowanie procesów ETL to finanse i bankowość. W tej branży ETL służy do konsolidacji danych z różnych źródeł, takich jak banki, firmy obsługujące karty kredytowe i instytucje finansowe. Dane te mogą być wykorzystywane do analizowania zachowań klientów, identyfikowania potencjalnych zagrożeń i wykrywania oszustw.

ETL jest również używany w branży medycznej do integracji danych ze źródeł, takich jak elektroniczna dokumentacja medyczna, roszczenia ubezpieczeniowe i badania kliniczne. Dane te można wykorzystać do monitorowania wyników pacjentów, analizowania skuteczności leczenia i poprawy świadczenia opieki zdrowotnej.

Kolejnym obszarem jest sprzedaż. W branży detalicznej, ETL służy do konsolidacji danych ze źródeł takich jak systemy punktów sprzedaży, programów lojalnościowych dla klientów i mediów społecznościowych. Dane te można wykorzystać do analizy zachowań klientów, śledzenia trendów sprzedaży i optymalizacji zarządzania zapasami.

Marketing: W branży marketingowej, ETL służy do konsolidacji danych z kampanii reklamowych, ankiet konsumenckich i mediów społecznościowych. Dane te mogą być wykorzystywane do analizowania zachowań klientów, śledzenia wyników działań marketingowych i optymalizacji kampanii marketingowych.

Procesy produkcji: ETL jest używany w przemyśle wytwórczym do integracji danych ze źródeł, takich jak systemy produkcyjne, systemy łańcucha dostaw   
i systemy kontroli jakości. Dane te można wykorzystać do monitorowania wydajności produkcji, identyfikowania wąskich gardeł i optymalizacji procesów produkcyjnych.

Oprócz wyżej wymienionych, proces ETL może być używany w różnych obszarach biznesowych takich jak np. branża mieszkaniowa, której dotyczy niniejsza praca. Cechą wspólną firm korzystających z procesów ETL jest to że przetwarzają one duże ilości danych oraz przechowują je w hurtowniach danych.

Hurtownie danych odgrywają kluczową rolę w kontekście procesów ETL. Stanowią scentralizowane repozytoria, gdzie dane z różnych źródeł są przechowywane w zoptymalizowanym formacie odpowiednim do analizy i raportowania[[9]](#footnote-9).

Znaczenie hurtowni danych w procesach ETL można opisać w sposób następujący[[10]](#footnote-10):

1. Centralizacja danych: Hurtownia danych pełni rolę centralnego punktu zbierania danych z różnych źródeł. Proces ETL wykorzystuje mechanizmy do wydobycia danych z różnych systemów, takich jak bazy danych, pliki tekstowe, interfejsy API itp., a następnie ładuje je do hurtowni danych. Ta centralizacja umożliwia skonsolidowane zarządzanie danymi i analizę z jednego miejsca.
2. Integracja danych: Proces ETL obejmuje przekształcanie danych, w tym konwersję formatów, normalizację, oczyszczanie, agregację, usuwanie duplikatów i wiele innych operacji. Hurtownia danych zapewnia strukturę danych, która ułatwia integrację różnorodnych źródeł danych. Umożliwia standaryzację danych, co sprzyja efektywnej analizie.
3. Optymalizacja wydajności: Hurtownie danych są zaprojektowane w celu umożliwienia szybkiego dostępu i wydajnego przetwarzania danych. Procesy ETL mogą wykorzystywać techniki optymalizacji, takie jak indeksowanie, partycjonowanie, struktury przechowywania danych i agregację, aby poprawić wydajność przetwarzania. Dzięki temu analiza danych z hurtowni jest efektywna.
4. Dostęp do danych historycznych: Hurtownie danych przechowują dane historyczne, umożliwiając analizę i raportowanie oparte na wielu okresach. Procesy ETL mogą być zaprojektowane w taki sposób, aby ładować i aktualizować dane historyczne, co umożliwia analizę trendów, porównywanie wyników i prognozowanie.

Podsumowując, hurtownia danych odgrywa istotną rolę w procesach ETL poprzez scentralizowanie danych, integrację różnorodnych źródeł, optymalizację wydajności i umożliwienie dostępu do danych historycznych. Stanowi podstawę efektywnej analizy danych, raportowania i podejmowania decyzji.

## Web scraping jako sposób pozyskiwania danych

Każdy proces ETL rozpoczyna się od pozyskania danych z różnego rodzaju źródeł. Typy danych pochodzących z różnych źródeł opisane zostały w poprzednim paragrafie. W tym podrozdziale zostanie przedstawione pozyskiwanie danych ze stron internetowych.

Web scraping to technika używana do automatycznego wydobywania danych ze stron internetowych. Wyodrębnione dane można wykorzystywać do różnych celów, takich jak analiza danych czy uczenie maszynowe. Dane są wyodrębniane za pomocą programowanych narzędzi zwanych „scraperami”, które mogą przeszukiwać strony internetowe i przechowywać je w ustrukturyzowanym formacie, takim jak np. baza danych lub arkusz kalkulacyjny. „Skrobanie” sieci można wykonać przy użyciu różnych języków programowania i bibliotek, takich jak Python, R, Scrapy, Selenium, Requests czy BS4. Wykorzystanie tego typu narzędzi pobierania danych ze stron internetowych spowszedniało w ostatnich latach ze względu na rozwój Internetu i liczbę stron internetowych, z których można pobierać dane. Web scraping może być złożonym i trudnym zadaniem, ponieważ strony internetowe mogą mieć skomplikowane do odczytu struktury, a pobieranie danych może być utrudnione przez właścicieli witryn[[11]](#footnote-11).

Proces Web scrapowania składa się z dwóch etapów jakimi są pozyskanie oraz ich przetworzenia do pożądanej formy. Dane najczęściej są pozyskiwane w dwóch formach jakimi są json (ang. JavaScript Object Notation) [2] lub kod HTML (ang. HyperText Markup Language) [2] .

JSON to popularny format do wymiany danych, używany najczęściej przez REST API (ang. representational state transfer) (interfejsy programowania aplikacji). Json pozwala na łączenie dużych ilości danych w pojedynczym pakiecie tekstu i wysyłanie ich do innych serwisów. Każda wartość danych ma swoją nazwę, czyli klucz. W przypadku tego formatu danych pozyskanie danych jest stosunkowo proste i można je uzyskać za pomocą biblioteki Requests (przykład poniżej)[[12]](#footnote-12).

JSON posiada wspólne cechy ze słownik w języku Python jednak są to różne struktury danych. JSON jest formatem przechowywania danych, często używanym do wymiany danych między różnymi aplikacjami. Składa się z par klucz-wartość, gdzie klucze są zawsze w postaci ciągu znaków, a wartości mogą być różnych typów, takich jak liczby, ciągi znaków, tablice, obiekty JSON itp. Z kolei słownik w języku Python jest strukturą danych, która również składa się z par klucz-wartość. Klucze są unikalne i muszą być niezmienne. Podobnie jak w plikach JSON, mogą nimi być obiekty takie jak ciągi znaków, liczby całkowite lub krotki. Wartości w słowniku mogą być różnych typów, takich jak liczby, ciągi znaków, listy, krotki, słowniki itp. Obie struktury danych umożliwiają dostęp do wartości za pomocą kluczy i mogą być zagnieżdżone, co oznacza, że wartości mogą również być innymi słownikami lub obiektami JSON.

Różnica między nimi polega na tym, że JSON jest formatem przechowywania danych, który można zapisywać i odczytywać z plików, natomiast słownik w języku Python jest strukturą danych, która istnieje wewnątrz programu i jest częścią języka. Oba narzędzia są użyteczne w różnych kontekstach. JSON jest popularny w przetwarzaniu danych między różnymi systemami, a słowniki w języku Python są przydatne w manipulowaniu danymi wewnątrz programu, umożliwiając efektywne indeksowanie i dostęp do danych za pomocą kluczy.

Rysunek 2. Przykładowy kod pozyskania danych w formie json przy użyciu biblioteki Requests.

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Źródło:** Opracowanie własne.

W powyższym przykładzie najpierw definiowany jest punkt końcowy interfejsu API   
i parametry jako zmienne. Następnie używane zostają metody request.get() do wysłania żądania GET do API, przekazując adres URL punktu końcowego i parametry jako argumenty. Odpowiedź jest przechowana w zmiennej o nazwie response. Następnie sprawdzany jest kod statusu odpowiedzi za pomocą atrybutu response.status\_code. Jeśli kod statusu to 200, można założyć, że żądanie zakończyło się pomyślnie, więc dane są wyodrębniane z odpowiedzi za pomocą metody response.json(), która zwraca słownik Pythona. Następnie można manipulować danymi, na przykład wydrukować je w konsoli lub utworzyć z nich ramkę danych. Jeśli kod stanu nie jest równy 200, można założyć, że żądanie nie powiodło się, więc pokazany zostanie komunikat o błędzie przy użyciu atrybutu response.text, który zawiera informację o błędzie zwróconym przez interfejs API.

Kolejnym formatem jest kod HTML odgrywa kluczową rolę w scrapowaniu stron internetowych, ponieważ za jego pomocą budowana jest struktura oraz zawartość stron internetowych. HTML jest językiem znaczników używanym do tworzenia stron internetowych i składa się z zestawu znaczników definiujących elementy i zawartość strony internetowej, takie jak tekst, obrazy, łącza, formularze i inne. Aby zeskrobać dane ze strony internetowej, scrapper musi zrozumieć strukturę i układ kodu HTML danej witryny. Kod HTML strony internetowej zawiera informacje o strukturze i zawartości strony internetowej, takie jak położenie poszczególnych elementów, hierarchia dokumentu i relacje między różnymi elementami. Kod HTML jest formatem danych, który został użyty w projekcie. Poniżej znajduję się przykładowy kod, który pozwala na wydobycie potrzebnych danych z kodu HTML (Rysunek 3.) [[13]](#footnote-13).

W tym przykładzie najpierw importowane są biblioteki Requests i BeautifulSoup. Następnie określamy adres URL strony, z której chcemy pobrać dane, przypisując go do zmiennej url. Następnie zostaje wysyłane żądanie GET na adres URL za pomocą request.get() i odpowiedź serwera zostaje zachowana w zmiennej response. Następnie zostaje dokonana analiza zawartości odpwiedzi HTML za pomocą Beautiful Soup, tworząc obiekt BeautifulSoup z treścią odpowiedzi i określonym parserem. Na koniec zostaje wyświetlony tytuł strony za pomocą soup.title.text i wszystkie linki na stronie za pomocą pętli for, która przegląda wszystkie znaczniki <a> na stronie i drukuje wartość atrybutu href dla każdego znacznika za pomocą link.get('href').

Rysunek 3. Przykładowy kod pozyskania danych w formacie kodu HTML przy użyciu bibliotek BeatifulSoup i Requests.

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Źródło:** Opracowanie własne.

JSON i HTML to dwa różne formaty, które często występują w procesie web scrapingu. Chociaż oba służą do ekstrakcji informacji z witryn internetowych, mają różne zastosowania   
i struktury.

JSON jest powszechnie stosowany do przesyłania danych między serwerem a aplikacją internetową. JSON organizuje dane w strukturalny sposób, wykorzystując pary klucz-wartość. Ma hierarchiczną strukturę i często jest wykorzystywany w interfejsach API oraz do przechowywania danych.

Z kolei HTML to język znaczników używany do tworzenia stron internetowych   
i strukturyzacji ich zawartości. HTML używa znaczników do definiowania elementów i ich atrybutów, tworząc strukturę drzewiastą, która reprezentuje wizualny układ strony internetowej.

Głównym celem JSON jest wymiana i przechowywanie danych. Często jest używany w interfejsach API do przesyłania danych między serwerami a aplikacjami. JSON reprezentuje strukturalne dane, takie jak informacje o użytkownikach lub szczegóły produktów. Z kolei HTML głównie służy do definiowania struktury i zawartości stron internetowych. Określa, w jaki sposób elementy powinny być wyświetlane w przeglądarce internetowej, w tym tekst, obrazy, linki, tabele i formularze.

Jeśli chodzi o reprezentację treści, JSON jest formatem czytelnym dla maszyn. Przechowuje dane za pomocą par klucz-wartość, tablic i zagnieżdżonych struktur. JSON obsługuje różne typy danych, takie jak ciągi znaków, liczby, wartości logiczne, tablice   
i obiekty. Jest często używany do przesyłania złożonych struktur danych między różnymi systemami. Z kolei HTML reprezentuje treść i strukturę strony internetowej, zawierającą tekst, obrazy, linki i inne elementy. Zazwyczaj nie zawiera on surowych danych ani złożonych struktur. HTML ma na celu opisanie wizualnej prezentacji strony internetowej, a nie samej reprezentacji danych.

Proces ekstrakcji danych z formatu JSON polega na wysyłaniu żądań HTTP do konkretnych punktów końcowych API i pobieraniu odpowiedzi w formacie JSON. Następnie pobrane dane można analizować i modyfikować za pomocą języków programowania obsługujących JSON, takich jak Python lub JavaScript. Z kolei web scraping z wykorzystaniem HTML polega na pobieraniu źródłowego kodu HTML strony internetowej i parsowaniu go w celu wyodrębnienia istotnych informacji. Często wymaga to użycia bibliotek lub narzędzi do web scrapingu, parsowania struktury HTML i wybierania konkretnych elementów za pomocą selektorów CSS lub wyrażeń XPath.

Podsumowując, JSON jest głównie stosowany do wymiany i przechowywania danych, podczas gdy HTML służy do strukturyzacji zawartości stron internetowych. JSON jest powszechnie używany w interfejsach API i reprezentuje złożone struktury danych, podczas gdy HTML definiuje wizualny układ stron internetowych. Proces ekstrakcji danych z formatu JSON obejmuje wysyłanie żądań API, podczas gdy web scraping HTML polega na wyodrębnianiu informacji z kodu źródłowego HTML stron internetowych.

## Wizualizacja danych jako narzędzie do analiz informacji

Wizualizacja danych jest niezbędnym narzędziem do analizy danych w wielu dziedzinach biznesowych oraz naukowych. Pozwala na przedstawienie dużych i złożonych zbiorów danych w łatwym do zrozumienia formacie, pomagając decydentom w szybkim identyfikowaniu wzorców, trendów i spostrzeżeń, które mogą być trudne do zidentyfikowania za pomocą tradycyjnych metod analitycznych[[14]](#footnote-14).

Istnieje wiele typów technik wizualizacji danych, które są powszechnie stosowane do reprezentowania danych w zrozumiały sposób. Poniżej zostanie przedstawione kilka z nich.

Wykresy słupkowe służą do przedstawiania danych liczbowych za pomocą prostokątnych słupków, przy czym wysokość lub długość każdego słupka reprezentują wartość danych. Wykresów słupkowych można używać do porównywania różnych kategorii lub do śledzenia zmian w czasie[[15]](#footnote-15).

Rysunek 4. Przykładowy wykres słupkowy.

Obraz zawierający wykres

Opis wygenerowany automatycznie

**Źródło:** Opracowanie własne.

Wykres liniowy służy do wyświetlania wartości ilościowych w ciągłym przedziale lub okresie. Wykres ten jest najczęściej używany do przedstawiania trendów i analizowania zmian danych w czasie. Wykresy liniowe są rysowane przez wykreślenie punktów danych na kartezjańskiej siatce współrzędnych, a następnie połączenie linii między wszystkimi tymi punktami. Zazwyczaj oś Y ma wartość ilościową, podczas gdy oś X to skala czasu lub sekwencja interwałów. Wartości ujemne mogą być wyświetlane poniżej osi X. Kierunek linii na wykresie jest dobrą metaforą danych: nachylenie w górę wskazuje, gdzie wartości wzrosły, a nachylenie w dół wskazuje, gdzie wartości spadły. Podróż linii po wykresie może tworzyć wzorce, które ujawniają trendy w zbiorze danych[[16]](#footnote-16).

Rysunek 5. Przykładowy wykres liniowy.

Obraz zawierający wykres

Opis wygenerowany automatycznie

**Źródło:** Opracowanie własne.

Mapy cieplne wizualizują dane za pomocą różnic w kolorystyce. W przypadku formatu tabelarycznego są przydatne do analizy danych wielowymiarowych poprzez umieszczanie zmiennych w wierszach i kolumnach oraz kolorowanie komórek w tabeli. Mapy cieplne są dobre do pokazywania wariancji między wieloma zmiennymi, ujawniania wszelkich wzorców, wyświetlania, czy jakieś zmienne są do siebie podobne oraz do wykrywania, czy istnieją jakiekolwiek korelacje. Dane liczbowe wymagają skali kolorów, która łączy jeden kolor w drugi, aby przedstawić różnicę w wysokich i niskich wartościach[[17]](#footnote-17).

Rysunek 6. Przykładowy wykres cieplny.

Obraz zawierający stół

Opis wygenerowany automatycznie

**Źródło:** <https://datavizcatalogue.com/methods/heatmap.html>, [06-04-2022].

Wykres punktowy umieszcza punkty w systemie współrzędnych kartezjańskich, aby wyświetlić wszystkie wartości między dwiema zmiennymi. Za jego pomocą można wykryć czy istnieje związek lub korelacja między nimi. Rodzaj korelacji można zinterpretować na podstawie wzorców ujawnionych na wykresie rozrzutu. Są to: dodatnie (wartości rosną razem), ujemne (jedna wartość maleje wraz ze wzrostem drugiej) lub zerowa (brak korelacji). Siłę korelacji można określić na podstawie tego, jak blisko siebie rozmieszczone są punkty na wykresie[[18]](#footnote-18).

Rysunek 7. Przykładowe wykresy punktowe.

Obraz zawierający wykres

Opis wygenerowany automatycznie

**Źródło:** Opracowanie własne.

Mapy geograficzne to popularny rodzaj wizualizacji danych używany do przedstawiania danych na podstawie lokalizacji. Można ich używać do wizualizacji wzorców   
i trendów w różnych regionach, co ułatwia zrozumienie rozkładu danych w różnych obszarach. Mapy geograficzne są często używane w dziedzinach takich jak epidemiologia, biznes i nauki społeczne do identyfikowania wzorców i trendów, które mogą nie być oczywiste w danych tabelarycznych lub tekstowych. Jedną z kluczowych zalet map geograficznych jest to, że mogą one zapewniać wizualną reprezentację danych, która może być łatwo zrozumiana przez osoby nieobeznane ze statystycznym lub technicznym żargonem. Na przykład mapa przedstawiająca rozpowszechnienie określonej choroby w różnych województwach[[19]](#footnote-19).

Rysunek 8. Przykładowy wykres mapy.

Obraz zawierający mapa

Opis wygenerowany automatycznie

**Źródło:** <https://datavizcatalogue.com/methods/heatmap.html>, [06-04-2022].

Powyżej zostały przedstawione wizualizacje które między innymi zostały przedstawione w projekcie, jednak istnieje wiele innych rodzajów technik wizualizacji danych, a wybór której należy użyć zależy od rodzaju analizowanych danych i potrzeb analizy. Istotnym elementem wizualizacji jest również dobór narzędzi.

Obecnie osoby zajmujące się wizualizacją danych mają dostęp do wielu narzędzi, dzięki którym stosunkowo małym nakładem pracy mogą budować złożone wizualizację.   
W projekcie został użyty program Microsoft Power BI. Zostanie on opisany w dalszej części pracy. Jest on jednym z najbardziej popularnych programów tego typu na rynku. Innymi programami, które są powszechnie wykorzystywane są np.: Tabelau, Google Data Studio, Looker, Qlik oraz wiele innych. Narzędzia te cechują się wieloma wspólnymi aspektami takimi jak[[20]](#footnote-20):

* Łączenie i integracja danych: Wszystkie te narzędzia umożliwiają użytkownikom łączenie się z różnymi źródłami danych, takimi jak arkusze kalkulacyjne, bazy danych, usługi w chmurze i inne aplikacje, umożliwiając użytkownikom tworzenie wizualizacji i pulpitów nawigacyjnych przy użyciu najbardziej aktualnych informacji,
* Interfejs „drag and drop”: te narzędzia wykorzystują interfejs „drag and drop”, aby umożliwić użytkownikom szybkie i łatwe tworzenie wykresów, wykresów i innych wizualizacji,
* Możliwości dowolnych wizualizacji: narzędzia te oferują szeroki zakres opcji wizualizacji, aby umożliwić użytkownikom tworzenie znaczących i wnikliwych raportów i pulpitów nawigacyjnych. Obejmują one wykresy kołowe, wykresy słupkowe, wykresy liniowe, wykresy punktowe, mapy cieplne i inne,
* Współpraca i udostępnianie: Wszystkie te narzędzia umożliwiają użytkownikom udostępnianie raportów, wizualizacji i pulpitów nawigacyjnych innym członkom zespołu lub interesariuszom w celu współpracy i podejmowania decyzji,
* Personalizacja: te narzędzia oferują różne opcje dostosowywania, takie jak kolory, style, czcionki i inne, aby umożliwić użytkownikom tworzenie spersonalizowanych   
  i profesjonalnie wyglądających raportów i pulpitów nawigacyjnych,

Podsumowując, narzędzia i techniki do wizualizacji danych mają na celu pomóc organizacjom w przekształcaniu dużych i złożonych zbiorów danych w praktyczne spostrzeżenia i umożliwiać podejmowanie świadomych decyzji.

# Analiza serwisu otodom.pl jako źródło danych

Poniższy rozdział skupia się na opisie oraz analizie serwisu internetowego jakim jest otodom.pl. Stanowi on źródło danych opracowanego projektu. W pierwszym podrozdziale zostanie przedstawiona ogólna charakterystyka tejże platformy, czyli przedstawienie czym ona jest oraz do czego służy. W drugim podrozdziale zostaną przedstawione jej funkcjonalności wraz z danymi które można z niej pozyskać.

## Charakterystyka serwisu

Otodom.pl to jeden z największych i najpopularniejszych serwisów internetowych oferujących nieruchomości w Polsce. Serwis ten stanowi platformę dla sprzedających, wynajmujących oraz poszukujących mieszkań, domów, działek, lokali użytkowych oraz komercyjnych. Otodom.pl został założony w 2006 roku i od tamtej pory stał się jednym   
z najważniejszych graczy na rynku nieruchomości w Polsce[[21]](#footnote-21).

Serwis otodom.pl działa na zasadzie portalu ogłoszeniowego, na którym prywatni oraz profesjonalni użytkownicy mogą umieszczać ogłoszenia dotyczące nieruchomości. Ogłoszenia te są w pełni darmowe i w łatwy sposób można je przeglądać dzięki intuicyjnemu interfejsowi użytkownika. Dostępne są filtry wyszukiwania, które pozwalają na precyzyjne określenie kryteriów poszukiwanej nieruchomości, takich jak lokalizacja, metraż, liczba pokoi, cena czy typ nieruchomości[[22]](#footnote-22).

Otodom.pl jest obecnie najbardziej popularnym serwisem nieruchomości w Polsce. Według najnowszego raportu PBI Gemius, w samym tylko wrześniu (rok 2020) odwiedziło go ponad 4 miliony rzeczywistych użytkowników co przełożyło się na ponad 18 milionów wizyt. Serwis aktywnie współpracuje z 975 firmami deweloperskimi. Ponadto otodom.pl charakteryzuję się największą liczbą ofert mieszkań. Poniższy wykres przedstawia liczbę ofert w Warszawie, które oferuję konkurencja. Pod tym względem inne serwisu cechują się mniej okazałą ofertą[[23]](#footnote-23).

Wykres 1. Porównanie liczby ogłoszeń mieszkań w Warszawie.

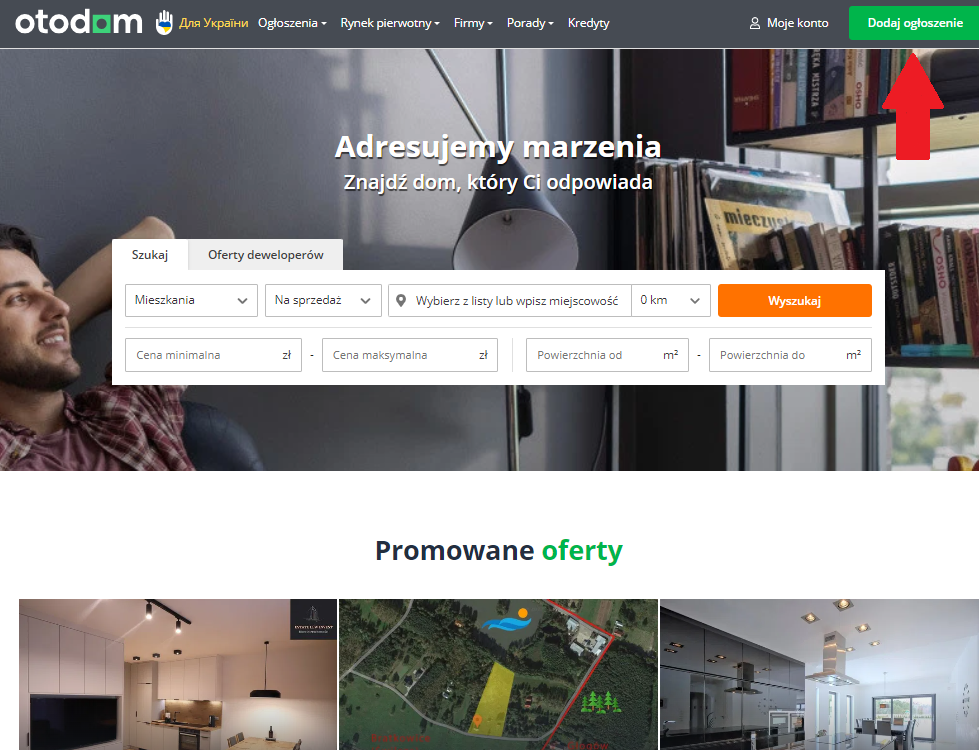
**Źródło**: Opracowanie własne na podstawie <https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/warszawa>, <https://warszawa.nieruchomosci-online.pl/mieszkania,sprzedaz/>, <https://mieszkanie.pl/ogloszenia/szukaj/mieszkanie/sprzedaz/warszawa>, <https://www.morizon.pl/mieszkania/warszawa/>, <https://gratka.pl/nieruchomosci/mieszkania/warszawa>, <https://www.domiporta.pl/mieszkanie/sprzedam?Localizations%5B0%5D.Name=warszawa>, [02-05-2023].

## Funkcjonalności oraz dane

Otodom.pl oferuje szereg funkcjonalności, które umożliwiają użytkownikom łatwe   
i efektywne korzystanie z serwisu. Główną funkcjonalnością na której bazuje serwis otodom.pl jest zamieszczanie ogłoszeń przez sprzedających. Proces ten jest stosunkowo intuicyjny dla użytkowników. Składa się on z trzech etapów, podczas których zbierane są dane o ofercie. Celem tego działania jest zbieranie parametrów, które pomagają w wyszukiwaniu jak najbardziej spersonalizowanych ogłoszeń. Etapami tymi są:

* Pierwszym etapem jest użycie przycisku “Dodaj ogłoszenie”, który widnieje w prawym górnym rogu strony głównej. Jest on w kolorze jasno zielonym, dzięki czemu wyróżnia się on na tle innych elementów na stronie, które są w odcieniach bardziej stonowanych. Jest to przedstawione na grafice poniżej (czerwona strzałka).

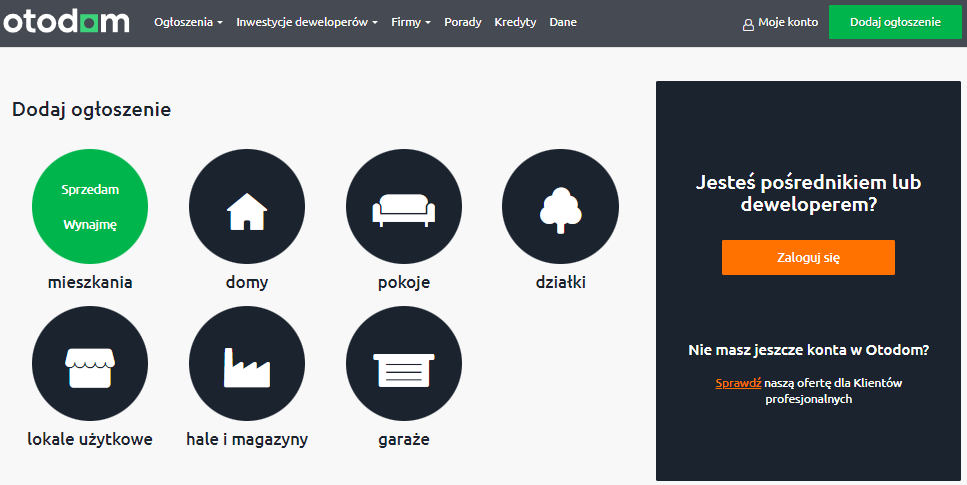
Rysunek 9. Strona główna serwisu otodom.



**Źródło**: <https://www.otodom.pl/>, [02-05-2023].

* Drugim etapem jest wybranie rodzaju ogłoszenia. Tutaj następuję pierwsze wprowadzenie danych przez sprzedającego. Użytkownik wybiera spośród takich możliwości jak: mieszkania, domy, pokoje, działki, lokale użytkowe, hale i magazyny oraz garaże. Każda z typu ofert może obrać dwa stany jakimi są “na sprzedaż” lub “na wynajem”. Widoczne to jest w grafice poniżej. W niniejszej pracy analizowane zostaną danę dotyczące ofert mieszkań na sprzedaż w warszawie więc dalsza ścieżka dodawania ogłoszenia będzie z tym zgodna.

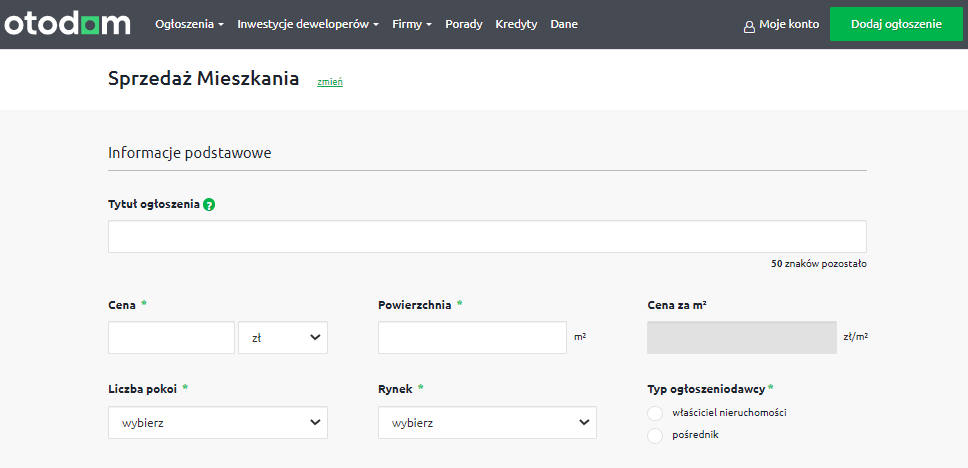
Rysunek 10. Strona z wyborem typu ogłoszenia serwisu otodom.pl.



**Źródło**: <https://www.otodom.pl/nowe-ogloszenie/kategoria/>, [02-05-2023].

* Podczas trzeciego etapu następuje wprowadzenie szczegółowych informacji   
  o nieruchomości. Są nimi między innymi powierzchnia, cena, liczba pokoi oraz wiele innych parametrów które zostaną omówione w dalszej części rozdziału. Wprowadzenie danych polega na uzupełnieniu formularza, którego część można zauważyć na poniższej grafice.

Rysunek 11. Przedstawienie formularza z wyborem typu ogłoszenia serwisu otodom.



**Źródło**: <https://www.otodom.pl/sprzedaz/mieszkanie/nowe-ogloszenie/>, [02-05-2023].

Następnie, na podstawie wprowadzonych danych, generowana jest strona   
z ogłoszeniem, która jest jednym z tysięcy źródeł danych opracowanego projektu. Dane, które są pozyskiwane dzielą się na cztery sekcję jakimi są:

* Lokalizacja, jest pozyskiwana z górnego sektora strony. Widnieją tam informację na temat województwa, miasta, dzielnicy, poddzielnicy oraz ulicy (Rysunek 13., oznaczenie **1**).
* Cena, jest umieszczona pod zdjęciami i wyrażona jest w złotówkach. Poniżej znajduję się również cena za metr kwadratowy. Jest to jedna z kluczowych parametrów każdej z ofert. (Rysunek 13., oznaczenie **2**).
* Dane szczegółowe, jest to sekcja w której zawiera się 10 parametrów oferty (Rysunek 13., oznaczenie **3**). Można tam znaleźć informację na temat:
* Powierzchni, wyrażonej w metrach w kwadratowych,
* Liczby pokoi,
* Piętra na którym znajduję się nieruchomość,
* Ceny czynszu,
* Formy własności,
* Stanie wykończenia,
* Informacji o balkonie/ogrodzie/tarasie.
* Miejsca parkingowego,
* Typu ogrzewania
* Informacji o obsłudze zdalnej
* Informację dodatkowe (Rysunek 12.) znajdują się w dolnych rejonach strony. Jest to sekcja w której znajdują się informacje takie jak:
* Rodzaj rynku (wtórny lub pierwotny),
* Typ ogłoszeniodawcy
* Data od której nieruchomość jest dostępna,
* Rok budowy,
* Rodzaj zabudowy,
* Rodzaj okien,
* Dostęp do windy,
* Media,
* Zabezpieczenia,
* Wyposażenie,
* Informacje dodatkowe,
* Materiał budynku.

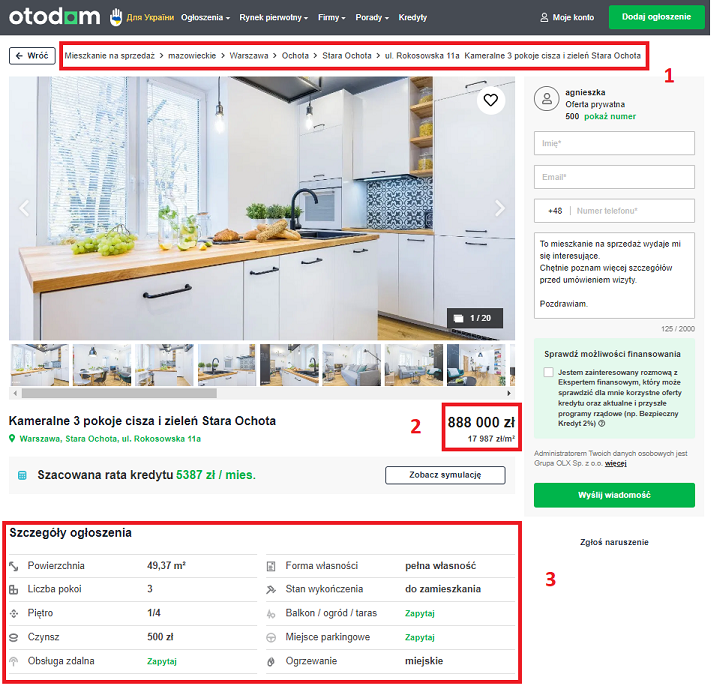
Rysunek 12. Element strony ogłoszenia z informacjami dodatkowymi.

Obraz zawierający tekst

Opis wygenerowany automatycznie

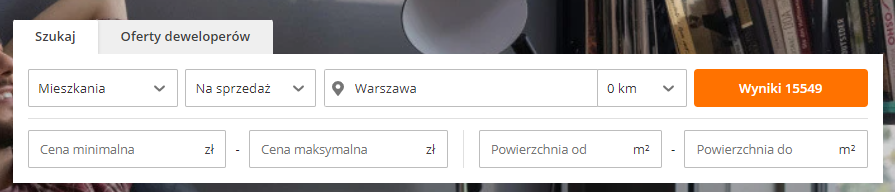
**Źródło**: <https://www.otodom.pl/pl/oferta/kameralne-3-pokoje-cisza-i-zielen-stara-ochota-ID4l96L>, [02-05-2023].

Rysunek 13. Przykładowa oferty z podziałem na sekcję danych.

**Źródło**: <https://www.otodom.pl/pl/oferta/kameralne-3-pokoje-cisza-i-zielen-stara-ochota-ID4l96L>, [02-05-2023].

Na podstawie przedstawionych danych, które wprowadził sprzedający, kupujący mogą z łatwością znaleźć nieruchomości o wymaganych parametrach. Jest to możliwe również dzięki panelowi na stronie głównej (Rysunek 14.). Użytkownik może w prosty sposób uzupełnić formularz i w oparciu o wprowadzone dane, zaprezentowane będą oferty zgodne   
z jego oczekiwaniami. Jest to istotny element projektu, ponieważ dzięki tej funkcjonalności możliwe było wygenerowanie adresu url do stron z ogłoszeniami mieszkań na sprzedaż   
w Warszawie.

Rysunek 14. Panel ułatwiający szukanie pożądanych ogłoszeń.



**Źródło**: <https://www.otodom.pl/> , [02-05-2023].

Przedstawione funkcjonalności oraz dane, które posiada serwis otodom.pl sprawiają, że jest on w pewnym sensie obszerną „bazą danych”, o rynku nieruchomości w Polsce. Dzięki web scrapowaniu, można darmowo czerpać informacje, przetwarzać je oraz dokonywać analiz na ich podstawie. W kolejnym rozdziale zostanie szczegółowo opisany tenże proces jak   
i narzędzia oraz cały projekt.

# Projekt procesu ETL

Poniższy rozdział opisuje koncepcje, założenia a także narzędzia, które zostały użyte do wykonania projektu.

## Koncepcja i założenia

Celem projektu jest opracowanie procesu ETL (Extract, Transform, Load) umożliwiającego pobieranie, transformowanie, magazynowanie i wizualizację danych   
z serwisu otodom.pl. Projekt zakłada automatyczne pozyskiwanie informacji   
o nieruchomościach dostępnych na serwisie otodom.pl, przetwarzanie tych danych w celu uzyskania spójnej struktury oraz przechowywanie ich w dedykowanej bazie danych. Następnie, dane te będą wizualizowane w formie interaktywnych wykresów i dashboardów, umożliwiających analizę i eksploatację zgromadzonych informacji.

Projekt opiera się czterech głównych etapach jakimi są:

1. Pozyskanie danych dzięki technikom web scrapowania przy użyciu takich narzędzi jak Visual Studio Code, Airflow, język programowania Python oraz biblioteka Selenium.
2. Przekształcenie danych, dzięki wykorzystaniu języka Python oraz biblioteki Pandas.
3. Ładowania danych, dzięki któremu zdobyte dane będą transportowane do bazy danych. W przedstawionym projekcie została wykorzystana baza danych SQLite.
4. Wizualizacja danych, dzięki której można lepiej zrozumieć dane, została wykonana za pomocą programu Microsoft Power Bi.

Założeniami projektu są następujące:

* Pobieranie danych następuję w raz w tygodniu (sobota),
* Dane dotyczą mieszkań na sprzedaż wyłącznie z Warszawy.

## Narzędzia i technologie

#### Python

Python to wysokopoziomowy język programowania, który został stworzony przez Guido van Rossuma i po raz pierwszy został wydany w 1991 roku. Jest to język interpretowany, co oznacza, że kod tworzony w Pythonie jest tłumaczony na bieżąco w trakcie jego wykonywania, a nie kompilowany do kodu maszynowego przed uruchomieniem.

Python jest językiem dynamicznym, co oznacza, że nie trzeba deklarować typów zmiennych przed ich użyciem. W związku z tym jest elastyczny i umożliwia programistom pisanie bardziej zwięzłego i czytelnego kodu. Python obsługuje zarówno programowanie proceduralne, jak i obiektowe, a także posiada cechy programowania funkcyjnego.

Python jest znany ze swojej czytelnej i przejrzystej składni, która ma na celu zapewnienie czytelności kodu. Zamiast stosowania nawiasów klamrowych do bloków kodu, Python używa wcięć. To oznacza, że odpowiednie wcięcia są niezbędne do oznaczenia bloków kodu, co wpływa na czytelność i spójność programu.

Rysunek 15. Przykładowy kod w języku python.

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Język Python ma szerokie zastosowanie i jest wykorzystywany w wielu dziedzinach, takich jak web development, analiza danych, sztuczna inteligencja, uczenie maszynowe, automatyka, robotyka i wiele innych.

Python jest głównym narzędziem służącym do realizacji projektu. Jest on jednym   
z najbardziej popularnych technologii wykorzystywanych do web scrapowania[[24]](#footnote-24). Jest to spowodowane dostępem do wielu bibliotek, które znaczącą ułatwiają ten proces. Właśnie dlatego został on wybrany do stworzenia projektu.

Podsumowując, Python jest językiem programowania ogólnego przeznaczenia, który cechuje się czytelną składnią, szerokimi możliwościami zastosowania, bogatą pulą bibliotek   
i modułów.

#### Visual Code Studio (IDE)

Visual Studio Code (VS Code) to bezpłatne, środowisko programistyczne stworzone przez firmę Microsoft. Jest dostępne na platformy Windows, macOS i Linux. VS Code zostało zaprojektowane z myślą o zapewnieniu wygodnego i efektywnego środowiska programistycznego dla wielu języków programowania i technologii.

Visual Studio Code oferuje wiele funkcji i narzędzi, które pomagają programistom   
w ich codziennej pracy. Oferuje edytor z podświetlaniem składni i inteligentnym podpowiadaniem, które ułatwiają pisanie kodu. VS Code umożliwia także debugowanie kodu bezpośrednio w edytorze, co ułatwia identyfikację i naprawę błędów.

Rysunek 16. Widok edytora kodu w Visual Code Studio.

Obraz zawierający tekst, zrzut ekranu, monitor, elektronika

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Środowisko Visual Studio Code integruje się również z systemami kontroli wersji, takimi jak Git, umożliwiając programistom monitorowanie zmian w kodzie i zarządzanie gałęziami. Dzięki wbudowanemu terminalowi, użytkownicy mogą wykonywać polecenia systemowe i uruchamiać skrypty bez potrzeby opuszczania edytora.

Jedną z najważniejszych cech VS Code jest jego elastyczność i rozszerzalność poprzez bibliotekę dostępnych rozszerzeń. Programiści mogą dostosować środowisko za pomocą rozszerzeń, takich jak wsparcie dla konkretnych frameworków, narzędzia do analizy statycznej, formatowania kodu i inne.

Dodatkowo, VS Code oferuje funkcję synchronizacji ustawień, która umożliwia przenoszenie preferencji, ustawień, rozszerzeń i motywów na różne komputery, co ułatwia zmienianie środowiska pracy.

Przedstawione cechy i właściwości spowodowały, że to właśnie IDE jakim jest Visual Code Studio zostało wykorzystane do wykonania projektu.

#### Selenium

Selenium jest narzędziem do automatyzacji testów aplikacji internetowych, ale może być również używane do web scrapowania. Jest to otwarte narzędzie, które umożliwia interakcję z przeglądarkami internetowymi w sposób programatyczny. Selenium obsługuje wiele przeglądarek, takich jak Chrome, Firefox, Safari, Edge, co daje użytkownikom szeroki zakres możliwości w automatyzacji działań w przeglądarce.

W przypadku web scrapowania, Selenium może być używane do przechodzenia przez strony internetowe, wypełniania formularzy, klikania w elementy, pobierania danych, a także do zarządzania plikami cookies i sesjami użytkownika. Działa to poprzez symulowanie akcji użytkownika w przeglądarce, umożliwiając programiście przechwytywanie informacji   
i manipulowanie stroną internetową.

Główną rolą Selenium w web scrapowaniu jest możliwość interakcji z dynamicznymi stronami internetowymi. Wiele nowoczesnych stron internetowych korzysta z technologii takich jak JavaScript, które dynamicznie aktualizują zawartość strony lub generują ją na podstawie interakcji użytkownika. Tradycyjne metody web scrapowania, takie jak pobieranie źródła HTML, mogą nie uwzględniać tych dynamicznych zmian. Selenium pozwala na symulację interakcji użytkownika, dzięki czemu możliwe jest pobieranie aktualnych danych ze stron internetowych, które wykorzystują technologie oparte o JavaScript.

Dodatkowo, Selenium oferuje również funkcje takie jak wyszukiwanie elementów na stronie za pomocą selektorów CSS lub XPath, co ułatwia znajdowanie konkretnych danych do scrapowania. Możliwość zarządzania sesjami użytkownika i cookies jest również przydatna   
w przypadku stron, które wymagają logowania lub utrzymywania stanu sesji.

Ważnym aspektem web scrapowania za pomocą Selenium jest to, że symuluje on rzeczywiste zachowanie użytkownika w przeglądarce internetowej. Może to pomóc   
w uniknięciu mechanizmów obronnych stosowanych przez strony internetowe, które próbują blokować lub utrudniać dostęp do danych. Jednak warto pamiętać, że niektóre strony internetowe mogą posiadać zabezpieczenia przeciwko automatycznemu scrapowaniu, takie jak Captcha, co może utrudnić lub uniemożliwić użycie Selenium w takich przypadkach.

Podsumowując, Selenium jest narzędziem do automatyzacji testów aplikacji internetowych, które można również wykorzystać do web scrapowania. Zapewnia możliwość interakcji z dynamicznymi stronami internetowymi i symuluje zachowanie użytkownika   
w przeglądarce, co umożliwia pobieranie aktualnych danych i manipulowanie stroną za pomocą kodu dlatego też zostało one wybrane do wykonania projektu.

#### SQLite

SQLite to silnik baz danych SQL. Jest to oprogramowanie o otwartym kodzie źródłowym, które umożliwia przechowywanie i zarządzanie danymi w formie baz danych. Co wyróżnia SQLite spośród innych systemów zarządzania bazami danych, to fakt, że baza danych SQLite jest przechowywana jako pojedynczy plik, co sprawia, że jest łatwa w implementacji   
i nie wymaga konfiguracji serwera czy instalacji odrębnych komponentów.

SQLite obsługuje standardowy język zapytań SQL (Structured Query Language), co czyni go zgodnym z większością systemów bazodanowych. Pozwala na tworzenie tabel, wstawianie, aktualizację, usuwanie i wyszukiwanie danych za pomocą zapytań SQL. Baza danych SQLite obsługuje również wiele typów danych, takich jak tekst, liczby, daty, wartości logiczne, co pozwala na przechowywanie różnorodnych informacji.

Ze względu na swoje cechy, takie jak łatwość użycia, przenośność, niezawodność   
i efektywność, SQLite jest szeroko stosowane w różnych aplikacjach, takich jak systemy zarządzania treścią, aplikacje mobilne, systemy wbudowane, narzędzia analityczne i wiele innych. Z tychże powodów silnik ten został wybrany do stworzenia projektu.

#### Apache Airflow

Apache Airflow to otwarte narzędzie do zarządzania przepływami pracy   
i harmonogramów, które pomaga w przetwarzaniu danych w sposób efektywny i automatyczny. Airflow został zaprojektowany z myślą o pracy z procesami ETL (Extract, Transform, Load) oraz innymi procesami przetwarzania danych, takimi jak przetwarzanie strumieniowe, uczenie maszynowe i analiza danych.

Airflow działa na zasadzie definiowania zadań i procesów za pomocą kodu   
w Pythonie, a następnie harmonogramowania ich do wykonania w odpowiednim czasie. Wszystkie zadania są zapisywane w formie skryptów w języku Python, co umożliwia programistom łatwe dodawanie niestandardowych zadań do procesów.

Airflow posiada pojęcie DAG (Directed Acyclic Graph), czyli skierowanego acyklicznego grafu. DAG jest grafem skierowanym, w którym każde zadanie jest reprezentowane przez węzeł, a zależności między zadaniami są reprezentowane przez krawędzie. DAG określa kolejność wykonywania zadań i zależności między nimi.

Airflow oferuje również wbudowany interfejs użytkownika, który umożliwia wizualizację postępów pracy i monitorowanie wykonywania zadań. Interfejs użytkownika Airflow umożliwia również wykonywanie ręczne poszczególnych zadań oraz ich ponowne uruchomienie w przypadku błędów (Rysunek 16.).

Airflow jest łatwo konfigurowalny i skalowalny. Można go dostosować do różnych potrzeb i wymagań, a także zintegrować z innymi narzędziami do przetwarzania danych, takimi jak Hadoop, Spark, SQL, NoSQL i wiele innych.

Podsumowując, Apache Airflow działa na zasadzie definiowania zadań i procesów za pomocą kodu w języku Python, a następnie harmonogramowania ich do wykonania   
w odpowiednim czasie. Airflow korzysta z DAG-ów, aby określić kolejność wykonywania zadań i ich zależności między nimi. Użytkownik Airflow może zarządzać harmonogramem wykonywania zadań, a wbudowany interfejs użytkownika umożliwia monitorowanie postępów pracy i ręczne wykonywanie zadań.

Rysunek 17. Widok interfejsu Apache Airflow.

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Źródło**: <https://github.com/apache/airflow>, [06-05-2023]

#### Power BI

Power BI to program do analizy i wizualizacji danych opracowany przez firmę Microsoft. Jest to narzędzie, które umożliwia użytkownikom biznesowym w łatwy i intuicyjny sposób tworzenie interaktywnych raportów, dashboardów oraz analizy danych.

Power BI składa się z kilku elementów, w tym:

* Power BI Desktop - jest to aplikacja desktopowa, która pozwala na tworzenie raportów i wizualizacji. Użytkownicy mogą importować dane z różnych źródeł, łączyć je   
  i przetwarzać w celu stworzenia analiz i raportów (Rysunek 18.)
* Power BI Service - jest to platforma w chmurze, która umożliwia udostępnianie raportów i dashboardów w organizacji. Użytkownicy mogą udostępniać swoje raporty innym użytkownikom, tworzyć grupy robocze oraz kontrolować dostęp do danych.
* Power BI Mobile - jest to aplikacja mobilna, która pozwala na przeglądanie   
  i interaktywne korzystanie z raportów i dashboardów na urządzeniach mobilnych.

Rysunek 18. Widok raportu stworzonego w Power BI.

Obraz zawierający tekst, zrzut ekranu, Czcionka, diagram

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Power BI oferuje wiele narzędzi i funkcjonalności, takich jak:

* Wizualizacje danych - Power BI oferuje wiele różnych wizualizacji, takich jak wykresy, tabele, mapy, diagramy i wiele innych. Użytkownicy mogą dostosowywać wizualizacje, aby lepiej oddawały ich potrzeby biznesowe,
* Pozyskiwanie danych – Power BI umożliwia pozyskiwanie danych z różnych źródeł takich jak pliki xlsx, json, hurtownie danych, data lake, strony internetowe , pliki csv,
* Transformacje danych - użytkownicy mogą łączyć, przetwarzać i korygować dane, aby uzyskać dokładne i spójne wyniki. Jest to możliwe również dzięki dodatkowi Power Querry,
* Automatyzacja raportów - Power BI pozwala na automatyczne generowanie raportów   
  i wysyłanie ich do wybranych użytkowników,
* Integracja z innymi narzędziami - Power BI integruje się z innymi narzędziami firmy Microsoft, takimi jak Excel, SharePoint i Teams, co umożliwia łatwe udostępnianie danych i raportów.

Power BI to wszechstronne narzędzie, które pozwala na szybkie i skuteczne analizowanie danych oraz prezentowanie wyników w formie interaktywnych raportów   
i dashboardów. Jest to popularne narzędzie wśród użytkowników biznesowych i analityków danych, którzy potrzebują szybkiego i łatwego sposobu na analizowanie danych[[25]](#footnote-25). Wymieniona wcześniej lista funkcjonalności oraz doświadczenie autora w wykorzystaniu tego narzędzia zadecydowało, że to właśnie Power BI został wybrany do wizualizacji pozyskanych danych   
w opracowanym projekcie.

# Implementacja

Poniższy rozdział skupia się na implementacji projektu. Zostanie w nim przedstawiony i opisany każdy jego etap poprzez prezentacje stworzonych rozwiązań przy użyciu wcześniej opisanych narzędzi.

## Pozyskanie danych

Proces pozyskania danych o ofertach mieszkań w warszawie z serwisu otodom.pl składa się z podprocesów, które zostaną poniżej opisane. Wynika to między innymi ze struktury serwisu, który charakteryzuję się paginacją stron oraz koniecznością użycia odpowiednich filtrów w panelu wyszukiwania.

Pierwszym etapem jest pozyskanie „startowego” adresu url, który pozwala na zawężenie ofert tylko do takich, które są celem analizy. Jest to możliwe dzięki panelowi na stronie głównej, który został przedstawiony we wcześniejszym fragmencie pracy. W celu pozyskania tegoż adresu, zostają wprowadzone konieczne parametry takie jak rodzaj nieruchomości („mieszkania”), rodzaj oferty („na sprzedaż”) oraz lokalizacja („Warszawa”). Jest to widoczne na poniższej grafice.

Rysunek 19. Panel wyszukiwarki z uzupełnionymi parametrami w serwisie otodom.pl

Obraz zawierający tekst, zrzut ekranu, Ludzka twarz, ubrania

Opis wygenerowany automatycznie

**Źródło**: <https://www.otodom.pl/>, [20-05-2023].

Po zatwierdzeniu parametrów, serwis przenosi użytkownika do pierwszej z wielu podstron z ofertami, której adres url jest wcześniej wymienionym „startowym”. Jak widać na poniższej grafice, otodom.pl stosuję technikę paginacji stron. Jest to istotna kwestia, która będzie uwzględniana w dalszych pracach.

Rysunek 20. Widok serwisu otodom.pl po zatwierdzeniu wymaganych parametrów.

Obraz zawierający tekst, zrzut ekranu, Strona internetowa, oprogramowanie

Opis wygenerowany automatycznie

**Źródło**:<https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/warszawa?distanceRadius=0&locations=%5Bcities_6-26%5D&viewType=listing> , [20-05-2023].

Kolejnym etapem procesu, który jest konieczny aby skorzystać z pozyskanego adresu url, jest stworzenie obiektu w języku Python, który będzie pełnił rolę przeglądarki. Jest to możliwe dzięki Selenium. Poniżej znajduję się grafika z kodem, który za to odpowiada.

Rysunek 21. Tworzenie obiektu przeglądarki.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Ten kod definiuje funkcję „configure\_driver()”, która konfiguruje sterownik przeglądarki Chrome. Funkcja tworzy obiekt „chrome\_options” reprezentujący opcje konfiguracyjne dla przeglądarki. Następnie dodaje trzy argumenty do chrome\_options: „--headless” (tryb bezinterakcyjny), „--no-sandbox” (tryb bezpieczny) i „--disable-dev-shm-usage” (mniejsze użycie pamięci dzielonej).

Kolejnym krokiem jest utworzenie obiektu „driver” klasy „webdriver.Chrome”, który reprezentuje sterownik przeglądarki Chrome. Sterownik ten jest skonfigurowany z użyciem ścieżki do pliku wykonywalnego „chromedriver.exe” oraz opcji „chrome\_options”.

Na koniec, zmienna „driver” otrzymuje wartość zwróconą przez funkcję „configure\_driver()”, co oznacza, że jest gotowa do użycia jako sterownik do przeprowadzania działań w przeglądarce Chrome.

Aby pozyskać adresy url z każdej z ofert, należy pozyskać informację o ich liczbie. Jest to możliwe dzięki temu, że jak wcześniej zostało wspomniane, strony te objęte są paginacją. Poniższy kod przedstawia pozyskanie informacji o liczbie stron z ofertami.

Rysunek 22. Pozyskanie informacji o liczbie stron z ofertami.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Powyższy kod, najpierw otwiera przeglądarkę i wczytuje stronę o podanym „linku startowym”. Następnie przewija stronę do końca za pomocą skryptu JavaScript, aby załadować całą treść. Tworzy obiekt BeautifulSoup, który analizuje źródło strony, przechowywane   
w „driver.page\_source”. To pozwala na ekstrakcję danych ze strony. Potem znajduje wszystkie elementy na stronie, które mają określoną klasę, i zapisuje je w liście „liczby stron”. Następnie pobiera tekst z czwartego elementu z listy „liczby\_stron” i przypisuje go do zmiennej „liczba\_stron” w której zawarta jest liczba wszystkich stron z ofertami.

Kolejnym etapem jest pozyskania adresów url do każdej z ofert z każdej strony   
z ofertami. Aby było to możliwe, należy stworzyć adresy do wszystkich ze stron z ofertami, wykorzystując informacje o ich liczbie oraz to że otodom.pl posługuje się paginacją stron. Polega to na modyfikacji adresu „startowego” poprzez zmianę ostatniego parametru jakim jest „page”, który na pierwszej stronie to 1. Widać to na poniższej grafice.

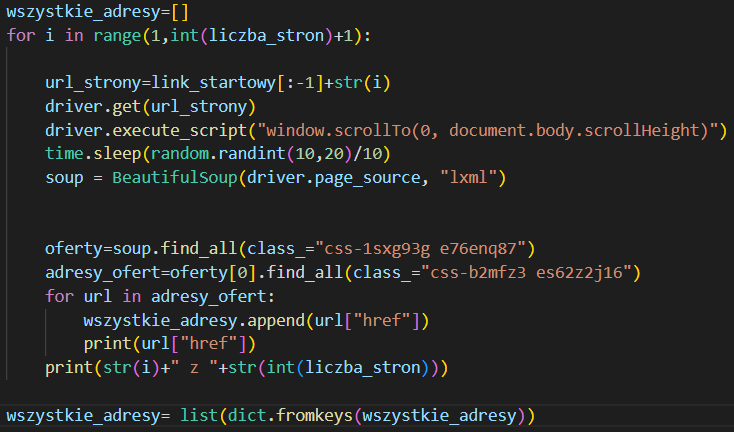
Rysunek 23. Przedstawienie budowy „startowego” adresu url.



**Źródło**:<https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/warszawa?distanceRadius=0&locations=%5Bcities_6-26%5D&viewType=listing> , [20-05-2023].

Modyfikacja oraz pozyskanie adresu url do każdej ze strony z ofertą mieszkania jest wykonywane za pomocą kodu przedstawionego na Rysunku 24.

Rysunek 24. Pozyskanie adresów url do wszystkich ofert.



**Źródło**: Opracowanie własne.

Na początku tworzona jest pusta lista o nazwie "wszystkie\_adresy". Następnie jest wykonywana pętla "for" iterująca po zakresie od 1 do „liczba\_stron” + 1, gdzie „liczba\_stron” jest konwertowana ze zmiennej typu string na liczbę całkowitą. Wewnątrz pętli:

1. Tworzony jest link poprzez łączenie link\_startowy (z usuniętym ostatnim znakiem)   
   i wartości "i" przekonwertowanej na ciąg znaków,
2. Przeglądarka otwiera ten link,
3. Wykonywany jest skrypt JavaScript, który przewija stronę do końca,
4. Program czeka losową ilość czasu między 1 a 2 sekundami w celu braku wykrycia, co skutkowałoby zablokowaniem adresu IP,
5. Tworzony jest obiekt BeautifulSoup z użyciem źródła strony przeglądarki,
6. Znajdowane są elementy o klasie "css-1sxg93g e76enq87", reprezentujące oferty,
7. Znajdowane są adresy ofert wewnątrz tych elementów,
8. Każdy znaleziony adres oferty jest dodawany do listy "wszystkie\_adresy"   
   i wyświetlany,
9. Wyświetlany jest komunikat informujący o postępie pętli.

Poza pętlą jest wykonywana operacja usuwania duplikatów z listy, ponieważ niektóre oferty widnieją na więcej niż jednej stronie, z tego powodu, że są ofertami promowanymi.

Ostatnim procesem w tej części projektu jest pozyskanie kodu html z każdej ze stron z ofertą i zapisanie jej w liście w celu przetworzenia danych w kolejnym etapie.

Rysunek 25. Pozyskanie kodu html z każdej ze stron z ofertą.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Początkowo tworzona jest pusta lista o nazwie "kod\_stron". Następnie jest wykonywana pętla "for", która iteruje przez elementy listy "wszystkie\_adresy". Wewnątrz pętli:

1. Tworzony jest pełny adres oferty poprzez dodanie ciągu znaków "https://www.otodom.pl" do wartości "i",
2. Przeglądarka otwiera ten adres oferty.
3. Program czeka losową ilość czasu między 1 a 2 sekundami.
4. Tworzony jest obiekt BeautifulSoup z użyciem źródła strony przeglądarki.
5. Obiekt BeautifulSoup reprezentujący kod strony jest dodawany do listy "kod\_stron".

## Transformacja danych

Celem tego etapu projektu jest przetworzenie pozyskanych wcześniej kodów html, które zawierają potrzebne informację oraz stworzenie z nich usystematyzowanej struktury   
w postaci tabeli

Dane będą pozyskiwane z czterech różnych sektorów strony. Będą to dane podstawowe, szczegółowe, lokalizacyjne oraz ceny. Z tego powodu tworzone są cztery puste ramki danych, których zadaniem jest przechowywanie tychże informacji.

Rysunek 26. Tworzenie pustych ramek danych.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Następnym etapem jest ich zapełnienie poprzez odpowiednie parsowanie kodu HTML, które jest widoczne na grafice poniżej.

Rysunek 27. Pozyskiwanie danych z kodu HTML.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

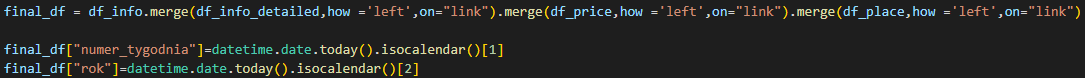
Dla każdego obiektu BeautifulSoup "soup" w liście "kod\_stron":

1. Znajdowane są elementy o klasie "css-wj4wb2 emxfhao1" reprezentujące tabelę informacji podstawowych.
2. Tworzone są puste listy "values" i "columns".
3. Dla każdego elementu w tabeli informacji:
   1. Znajdowane są elementy wewnątrz tego elementu o klasie "css-1ccovha estckra9".
   2. Dla każdego znalezionego elementu:
      1. Znajdowane są elementy wewnątrz niego o klasie "css-1qzszy5 estckra8".
      2. Pierwszy element z tych znalezionych elementów jest dodawany do listy "columns".
      3. Ostatni element z tych znalezionych elementów jest dodawany do listy "values".
4. Tworzony jest obiekt DataFrame z listy "values" i następnie jest transponowany.
5. Kolumny DataFrame są ustawiane na wartości z listy "columns".
6. Dodawana jest nowa kolumna "link" zawierająca wartości z listy "wszystkie\_adresy".
7. DataFrame "table\_info" jest dołączany do DataFrame "df\_info" za pomocą funkcji concat.

Analogicznie uzupełniane są ramki danych odpowiadające za dane szczegółowe, dane lokalizacyjne oraz ceny.

Kolejnym etapem jest połączenie ze sobą zapełnionych ramek danych w celu łatwiejszego importowania danych do bazy danych oraz dodanie zmiennych mówiących o roku oraz numerze tygodnia pomiaru. Jest to wykonywane poniższym kodem.

Rysunek 28. Łączenie ramek danych oraz dodanie zmiennych czasowych.



**Źródło**: Opracowanie własne.

Najpierw tworzony jest nowy DataFrame o nazwie "final\_df", który powstaje   
w wyniku połączenia DataFrame'ów "df\_info", "df\_info\_detailed", "df\_price" i "df\_place" za pomocą funkcji „merge”. Połączenie odbywa się przy użyciu kolumny "link" z każdego DataFrame. Następnie do DataFrame "final\_df" dodawana są dwie kolumny o nazwie "numer\_tygodnia" oraz „rok”, które zawierają numer tygodnia oraz rok z daty pomiaru za pomocą metody isocalendar() dla obiektu datetime.date.today().

## Ładowanie danych

Celem kolejnego etapu projektu jest ładowanie danych z utworzonej wcześniej ramki danych do bazy danych sqlite3.

Pierwszym etapem jest stworzenie bazy danych. Jest to wykonywane w następujący sposób widoczny na Rysunku 27.

Rysunek 29. Tworzenie bazy danych.

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Najpierw tworzony jest obiekt silnika („engine”) przy użyciu funkcji create\_engine() z biblioteki sqlalchemy. Silnik ten jest używany do komunikacji z bazą danych SQLite o nazwie "mieszkania\_warszawa.sqlite". Następnie tworzone zostaje połączenie („conn”) z bazą danych za pomocą funkcji connect() z biblioteki sqlite3. To połączenie będzie używane do wykonywania operacji na bazie danych. Kolejnym etapem jest tworzenie kursora („cursor”) za pomocą metody cursor() na obiekcie połączenia („conn”). Kursor ten będzie używany do wykonywania poleceń SQL na bazie danych poprzez wywołanie odpowiednich metod. Potem definiowany zostaję ciąg znaków „sql\_query”, który zawiera polecenie SQL do utworzenia tabeli "mieszkania\_magisterka". Kolumny tabeli oraz ich typy danych są określone   
w poleceniu. Ostatnim procesem jest wykonanie polecenie SQL zawarte w zmiennej „sql\_query” przy użyciu metody execute() na kursorze („cursor”). Polecenie tworzy tabelę "mieszkania\_magisterka" w bazie danych SQLite, jeśli tabela jeszcze nie istnieje.

Wszystkie zmienne są danymi typu string. Jest to spowodowane tym, że zamiana na pożądane typy zmiennych będzie przeprowadzona podczas wizualizacji w programie Power BI. Wynika to z tego, że wymaga to mniejszego nakładu pracy niż „czyszczenie”   
i formatowanie danych w języku Python.

Ostatnim elementem tego etapu projektu jest załadowanie pozyskanych wcześniej danych do stworzonej bazy danych. Jest to wykonywane za pomocą kodu widocznego na Rysunku 28.

Rysunek 30. Ładowanie danych do bazy danych.



**Źródło**: Opracowanie własne.

Powyższy kod wywołuję metodę to\_sql() na obiekcie DataFrame „final\_df”. Metoda ta służy do zapisywania danych z DataFrame do bazy danych. Parametry metody to:

* "mieszkania\_warszawa": Nazwa tabeli, do której mają zostać zapisane dane.
* „engine”: Obiekt silnika („engine”), który reprezentuje połączenie z bazą danych.
* „index=False”: Opcjonalny parametr, który określa, czy indeksy DataFrame mają być zapisane jako osobna kolumna w tabeli. W tym przypadku wartość False oznacza, że indeksy nie będą zapisywane.
* „if\_exists='append’”: Opcjonalny parametr, który określa, co zrobić, jeśli tabela już istnieje. Wartość 'append' oznacza, że nowe dane zostaną dołączone do istniejącej tabeli.

Po tym następuje zamknięcie połączenia z bazą danych SQLite poprzez wywołanie metody „close()” na obiekcie połączenia („conn”).

Dzięki opisanym czynnościom, DataFrame „final\_df” zostanie zapisany do tabeli "mieszkania\_warszawa" w bazie danych SQLite, a następnie połączenie z bazą danych zostanie zamknięte. Dane będą dostępne do dalszego użytku w bazie danych.

## Automatyzacja procesu

Automatyzacja procesu etl jest wykonywana dzięki narzędziu Apache Air Flow. Polega to na zbudowaniu pliku DAG, który jest odpowiedzialny za wykonywanie kodu   
w ustalonym interwale czasowym. Poniżej znajduję się opis każdego etapu opracowania DAG’a, który odpowiada za cykliczne wykonywanie procesu pozyskania, transformacji oraz ładowania scrapowanych danych z serwisu otodom.pl.

Pierwszym etapem jest import funkcji, która odpowiada za cały proces etl. Jest to potrzebne aby dag miał informację jaki kod ma być uruchamiany.

Rysunek 31. Importowanie funkcji.



**Źródło**: Opracowanie własne.

Drugim etapem jest tworzenie słownika „default\_args”, który zawiera argumenty domyślne dla zadań w DAG. W tym przypadku właścicielem jest "bartek", zależności na przeszłe zadania są wyłączone („depends\_on\_past=False”), data rozpoczęcia to 4 marca 2023 roku („start\_date=datetime(2023, 4, 3)”), adres e-mail do powiadomień to 'bartoszradz@gmail.com.com', wyłącza powiadomienia e-mail w przypadku błędu („email\_on\_failure=False”) oraz podczas ponawiania („email\_on\_retry=False”), przewiduje jedno ponowienie („retries=1”), a czas oczekiwania między ponowieniami to jedna minuta („retry\_delay=timedelta(minutes=1)”). Jest to pokazane na Rysunku 30.

Rysunek 32. Tworzenie parametrów DAG’a.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Kolejnym etapem, jest stworzenie obieku „dag” z nazwą „otodom\_dag” i przypisane zostają mu, wcześniej wymienione „default\_args” jako argumenty domyślne. Dodany zostaje również opis oraz interwał jakim jest czas 7 dni. Widać to na rysunku poniżej.

Rysunek 33. Tworzenie obiektu DAG.

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Ostatnim procesem jest tworzenie obiektu „run\_etl” klasy „PythonOperator”, który reprezentuje wykonanie zadania ETL dla danych z serwisu otodom.pl. Przypisany zostaje identyfikator zadania jako „whole\_otodom\_etl”. Następnie wskazana zostaje funkcja „run\_otodom\_etl”, która ma zostać wykonana oraz wskazany jest dag, który został wcześniej stworzony. Kod jest widoczny na Rysunku 32.

Rysunek 34. Tworzenie obiektu run\_etl.

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

## Wizualizacja danych

Ostatnim elementem projektu jest wizualizacja pozyskanych danych w programie Power BI. Składa się ona z trzech sekcji które zostaną opisane poniżej.

Pierwszą stroną wizualizacji jest „Strona początkowa”, która wyświetla się jako pierwsza po wejściu do raportu. Jest ona przedstawiona na grafice poniżej. Jej zadaniem jest poinformowanie użytkownika o czym jest i co zawiera raport.

Rysunek 35. Strona startowa wizualizacji.

Obraz zawierający tekst, mapa, zrzut ekranu, diagram

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

W górnym sektorze widnieje tytuł wizualizacji, przyciski kierujące do dalszych stron oraz informacja o aktualnym numerze tygodnia. Natomiast w środkowej części strony umieszczony jest tekst zawierający informacje o raporcie, grafika przedstawiająca mapę warszawy w podziale na dzielnice oraz przycisk „Przejdź do raportu”, którego użycie skutkuje przeniesieniem do kolejnej strony raportu.

Kolejną stroną raportu jest zakładka o nazwie „Ceny i oferty”. Użytkownik jest na nią przenoszony po użyciu przycisku z takim samym tekstem jak nazwa strony. Jest to również możliwe przy użyciu przycisku „Przejdź do raportu” na stronie początkowej. Strona ta przedstawiona jest na grafice poniżej.

Rysunek 36. Strona „Ceny i oferty”.

Obraz zawierający tekst, numer, diagram, zrzut ekranu

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Powyższa strona podzielona jest na dwa obszary. Pierwszym z nich, który znajduję się po lewej stronie, są filtry danych. Ich celem jest umożliwienie użytkownikowi dokonania analizy nieruchomości o jak najbardziej szczegółowych parametrach. Są to między innymi filtry dotyczące ceny za metr kwadratowy, powierzchni, liczby pokoi i innych, które widoczne są na grafice powyżej.

Drugim obszarem jest sekcja poświęcona wykresom. Użytkownik może zaobserwować zmienność w czasie takich miar jak liczba ofert, mediana ceny za metr kwadratowy oraz mediana cen mieszkań.

Kolejną stroną dostępną stroną w raporcie jest strona do której użytkownik jest przenoszony po użyciu przycisku „Dzielnice”. Celem tej strony jest przedstawienie jak różnią się wcześniej wymienione trzy miary w zależności od dzielnicy Warszawy, w której zlokalizowane są nieruchomości. Strona „Dzielnice” przedstawiona jest na poniższej grafice.

Rysunek 37. Strona „Dzielnice”.

Obraz zawierający tekst, diagram, zrzut ekranu, numer

Opis wygenerowany automatycznie

**Źródło**: Opracowanie własne.

Tak samo jak poprzednio opisywana strona, ta również dzieli się dwie sekcje. Po lewej stronie znajduje się sekcja zawierająca filtry a po prawej stronie znajduje się wizualizacja. Jest ona w formie mapy Warszawy w podziale na dzielnice. Charakteryzuję się ona tym, że im większa wartość miary tym odcień obszaru jest ciemniejszy.

Dodatkowo wizualizacja ta posiada trzy przyciski, które umieszczone są nad mapą.   
W efekcie tego jaki przycisk zostanie użyty przez użytkownika, przedstawiona zostanie inna miara na wizualizacji. Miarami są mediana ceny ze metr kwadratowy, mediana ceny mieszkania oraz liczba ofert.

Podsumowując, przedstawione wizualizacje wykonane w programie Power BI umożliwiają użytkownikom dokonywania analiz warszawskiego rynku mieszkań w ujęciu tygodniowym oraz w podziale na dzielnice. Dzięki nim użytkownicy mogą podejmować decyzje biznesowe, oparte o przedstawione liczby.

Zakończenie

W niniejszej pracy magisterskiej udało się osiągnąć główny cel, którym było stworzenie kompletnego procesu ETL oraz wizualizacji danych dotyczących mieszkań na sprzedaż w Warszawie, pozyskiwanych z serwisu otodom.pl. Dzięki zastosowaniu techniki web scrapingu, udało się skutecznie pozyskać informacje z witryny otodom.pl i przekształcić je   
w odpowiedni format. Następnie dane zostały załadowane do bazy danych, co zapewniło łatwy dostęp do zgromadzonych informacji.

Ważnym elementem tej pracy było również opracowanie interaktywnego dashboardu, który umożliwia przejrzystą wizualizację danych. Dzięki różnorodnym wykresom, mapom   
i filtrom, użytkownicy mają możliwość eksplorowania danych, analizy trendów oraz wyciągania wniosków. Dashboard ten stanowi wartościowe narzędzie dla osób zainteresowanych rynkiem mieszkaniowym w Warszawie, umożliwiając im szybkie   
i efektywne zrozumienie zgromadzonych informacji.

Podczas realizacji pracy napotkano również pewne wyzwania związane z web scrapingiem oraz przetwarzaniem danych. Konieczne było opracowanie skutecznych strategii pobierania danych, a także zapewnienie spójności i czystości zgromadzonych informacji. Jednak dzięki starannemu planowaniu i podejściu iteracyjnemu, udało się skutecznie pokonać te trudności i osiągnąć zamierzone cele.

Ważnym aspektem tej pracy jest również możliwość dalszego rozwoju projektu. Istnieje wiele potencjalnych usprawnień i rozszerzeń, które mogą zostać wprowadzone   
w przyszłości. Przykładowo, można rozważyć dodanie dodatkowych funkcji do dashboardu, takich jak prognozowanie cen mieszkań na podstawie zebranych danych czy integrację   
z innymi źródłami danych z branży nieruchomości czy przeniesienia kodu do chmury.

Podsumowując, niniejsza praca magisterska przyczynia się do rozwoju dziedziny analizy danych w kontekście rynku nieruchomości, poprzez dostarczenie kompletnego procesu ETL oraz interaktywnego dashboardu. Otwiera ona możliwości dla analizy rynku mieszkaniowego w Warszawie, umożliwiając precyzyjne prognozowanie trendów oraz lepsze zrozumienie zachodzących zmian. Projekt może być użyteczny dla osób zainteresowanych rynkiem nieruchomości w stolicy Polski.

LITERATURA

**Książki**

1. P. Crickard, *Data Engineering with Python*, Packt, Birmingham 2020.
2. R. Mitchell, *Web Scraping with Python*, O’Reilly, Sebastopol 2015.
3. C. Nussbaumer Knafic, *Storytelling with Data*, John Wiley & Sons, 2015.
4. C. O. Wilke, *Podstawy wizualizacji danych. Zasady tworzenia atrakcyjnych wykresów*, Helion, Gliwice 2015.
5. A. Pelikant, Hurtownie danych. Od przetwarzania analitycznego do raportowania, Helion, Gliwice 2021.
6. J. Steele, N. Iliinsky, *Beautiful Visualization: Looking at Data through the Eyes of Experts (Theory in Practice)*, O'Reilly, 2010.
7. G. Baldowska, M. Gumowska, M. Ulatowska,  *Rynek nieruchomości mieszkaniowych w Warszawie i aglomeracji warszawskiej w I kwartale 2023 r.,* NBP, Warszawa, maj 2023.

**Strony internetowe**

1. Poksiński P., *PRINCE2*, materiał dostępny w wersji elektronicznej pod adresem internetowym: http://www.poksinski.com/pdf\_files/PRINCE2.pdf [18-09-2012]
2. *ETL overview,* <https://www.altexsoft.com/blog/etl-vs-elt/>, [19-03-2023].
3. *What is ETL?*, <https://www.ibm.com/topics/etl>, [19-03-2023].
4. *What is ETL (extract transform load)?*, <https://www.informatica.com/resources/articles/what-is-etl.html>, [19-03-2023].
5. <https://datavizcatalogue.com/methods/heatmap.html>, [06-04-2022].
6. *The Best Data Visualization Tools Of 2023*, <https://www.forbes.com/advisor/business/software/best-data-visualization-tools/>, [07-04-2023].
7. *Ranking portali sprzedających nieruchomości*, <https://ceo.com.pl/ranking-portali-sprzedajacych-nieruchomosci-53865#OTODOM> , [02-05-2023].
8. <https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/warszawa>, [02-05-2023].
9. <https://warszawa.nieruchomosci-online.pl/mieszkania,sprzedaz/>, [02-05-2023].
10. <https://mieszkanie.pl/ogloszenia/szukaj/mieszkanie/sprzedaz/warszawa>, [02-05-2023].
11. <https://www.morizon.pl/mieszkania/warszawa/>, [02-05-2023].
12. <https://gratka.pl/nieruchomosci/mieszkania/warszawa>, [02-05-2023].
13. <https://www.otodom.pl> , [02-05-2023].
14. <https://www.otodom.pl/nowe-ogloszenie/kategoria/>, [02-05-2023].
15. <https://www.otodom.pl/pl/oferta/kameralne-3-pokoje-cisza-i-zielen-stara-ochota-ID4l96L>, [02-05-2023].
16. *Best Programming Languages for Effective Web Scraping*, <https://oxylabs.io/blog/best-web-scraping-language>, [05-05-2023].
17. <https://github.com/apache/airflow>, [06-05-2023].
18. Power BI Vs Tableau: Difference and Comparison, <https://www.simplilearn.com/tutorials/power-bi-tutorial/power-bi-vs-tableau#features_of_power_bi> , [14-05-2023].

Spis wykresów

[Wykres 1. Porównanie liczby ogłoszeń mieszkań w Warszawie. 21](#_Toc136207758)

Spis rysunków

[Rysunek 1. Przedstawienie procesu ETL. 7](#_Toc136207548)

[Rysunek 2. Przykładowy kod pozyskania danych w formie json przy użyciu biblioteki Requests. 11](#_Toc136207549)

[Rysunek 3. Przykładowy kod pozyskania danych w formacie kodu HTML przy użyciu bibliotek BeatifulSoup i Requests. 13](#_Toc136207550)

[Rysunek 4. Przykładowy wykres słupkowy. 14](#_Toc136207551)

[Rysunek 5. Przykładowy wykres liniowy. 15](#_Toc136207552)

[Rysunek 6. Przykładowy wykres cieplny. 16](#_Toc136207553)

[Rysunek 7. Przykładowe wykresy punktowe. 17](#_Toc136207554)

[Rysunek 8. Przykładowy wykres mapy. 18](#_Toc136207555)

[Rysunek 9. Strona główna serwisu otodom. 22](#_Toc136207556)

[Rysunek 10. Strona z wyborem typu ogłoszenia serwisu otodom.pl. 23](#_Toc136207557)

[Rysunek 11. Przedstawienie formularza z wyborem typu ogłoszenia serwisu otodom. 23](#_Toc136207558)

[Rysunek 12. Element strony ogłoszenia z informacjami dodatkowymi. 25](#_Toc136207559)

[Rysunek 13. Przykładowa oferty z podziałem na sekcję danych. 26](#_Toc136207560)

[Rysunek 14. Panel ułatwiający szukanie pożądanych ogłoszeń. 27](#_Toc136207561)

[Rysunek 15. Przykładowy kod w języku python. 29](#_Toc136207562)

[Rysunek 16. Widok edytora kodu w Visual Code Studio. 30](#_Toc136207563)

[Rysunek 17. Widok interfejsu Apache Airflow. 34](#_Toc136207564)

[Rysunek 18. Widok raportu stworzonego w Power BI. 35](#_Toc136207565)

[Rysunek 19. Panel wyszukiwarki z uzupełnionymi parametrami w serwisie otodom.pl 37](#_Toc136207566)

[Rysunek 20. Widok serwisu otodom.pl po zatwierdzeniu wymaganych parametrów. 38](#_Toc136207567)

[Rysunek 21. Tworzenie obiektu przeglądarki. 38](#_Toc136207568)

[Rysunek 22. Pozyskanie informacji o liczbie stron z ofertami. 39](#_Toc136207569)

[Rysunek 23. Przedstawienie budowy „startowego” adresu url. 40](#_Toc136207570)

[Rysunek 24. Pozyskanie adresów url do wszystkich ofert. 40](#_Toc136207571)

[Rysunek 25. Pozyskanie kodu html z każdej ze stron z ofertą. 41](#_Toc136207572)

[Rysunek 26. Tworzenie pustych ramek danych. 42](#_Toc136207573)

[Rysunek 27. Pozyskiwanie danych z kodu HTML. 42](#_Toc136207574)

[Rysunek 28. Łączenie ramek danych oraz dodanie zmiennych czasowych. 43](#_Toc136207575)

[Rysunek 29. Tworzenie bazy danych. 44](#_Toc136207576)

[Rysunek 30. Ładowanie danych do bazy danych. 45](#_Toc136207577)

[Rysunek 31. Importowanie funkcji. 46](#_Toc136207578)

[Rysunek 32. Tworzenie parametrów DAG’a. 47](#_Toc136207579)

[Rysunek 33. Tworzenie obiektu DAG. 47](#_Toc136207580)

[Rysunek 34. Tworzenie obiektu run\_etl. 47](#_Toc136207581)

[Rysunek 35. Strona startowa wizualizacji. 48](#_Toc136207582)

[Rysunek 36. Strona „Ceny i oferty”. 49](#_Toc136207583)

[Rysunek 37. Strona „Dzielnice”. 50](#_Toc136207584)

Streszczenie

Praca magisterska skupia się na opracowaniu kompletnego procesu ETL oraz stworzeniu interaktywnego dashboardu służącego do wizualizacji danych dotyczących mieszkań na sprzedaż w Warszawie, pozyskiwanych z serwisu otodom.pl. W ramach pracy zastosowano technikę web scrapingu w celu pozyskania danych z serwisu oraz ich przekształcenia do odpowiedniego formatu. Następnie dane zostały załadowane do bazy danych, zapewniając łatwy dostęp do zgromadzonych informacji. Stworzony interaktywny dashboard umożliwia użytkownikom przejrzystą wizualizację danych za pomocą różnorodnych wykresów, map   
i filtrów, umożliwiając analizę trendów i eksplorację danych. Praca omawia również napotkane wyzwania związane z web scrapingiem i przetwarzaniem danych. Wyniki pracy mają praktyczne zastosowanie dla analizy rynku mieszkaniowego w Warszawie, dostarczając wartościowych narzędzi dla osób zainteresowanych tym rynkiem.

Summary

The master's thesis focuses on developing an ETL process and data visualization for apartment listings in Warsaw obtained through web scraping from the otodom.pl website. The study utilizes web scraping techniques to extract data from the website and transform it into the appropriate format. Subsequently, the data is loaded into a database, ensuring easy access to the collected information. An interactive dashboard has been created to provide users with clear data visualization, including various charts, maps, and filters, enabling trend analysis and data exploration. The thesis also addresses challenges associated with web scraping and data processing and proposes further project enhancements. The outcomes of this study have practical implications for the analysis of the housing market in Warsaw, offering valuable tools for individuals interested in this market.

1. G. Baldowska, M. Gumowska, M. Ulatowska,  *Rynek nieruchomości mieszkaniowych w Warszawie i aglomeracji warszawskiej w I kwartale 2023 r.,* NBP, Warszawa, maj 2023, str. 3. [↑](#footnote-ref-1)
2. Tamże. [↑](#footnote-ref-2)
3. What is ETL? <https://www.ibm.com/topics/etl> [19-03-2023]. [↑](#footnote-ref-3)
4. What is ETL? <https://www.ibm.com/topics/etl> [19-03-2023]. [↑](#footnote-ref-4)
5. What is ETL? <https://www.ibm.com/topics/etl> [19-03-2023]. [↑](#footnote-ref-5)
6. Tamże. [↑](#footnote-ref-6)
7. What is ETL (extract transform load)? <https://www.informatica.com/resources/articles/what-is-etl.html> [19-03-2023]. [↑](#footnote-ref-7)
8. P. Crickard, *Data Engineering with Python,* Packt, Birmingham 2020, str. 57. [↑](#footnote-ref-8)
9. A. Pelikant, *Hurtownie danych. Od przetwarzania analitycznego do raportowania*, Helion, Gliwice 2021, str. 23. [↑](#footnote-ref-9)
10. Tamże. [↑](#footnote-ref-10)
11. R. Mitchell, *Web Scraping with Python*, O’Reilly, Sebastopol 2015, str. 9. [↑](#footnote-ref-11)
12. R. Mitchell, *Web Scraping with Python*, O’Reilly, Sebastopol 2015, str. 23. [↑](#footnote-ref-12)
13. R. Mitchell, *Web Scraping with Python*, O’Reilly, Sebastopol 2015, str. 24. [↑](#footnote-ref-13)
14. C. O. Wilke, *Podstawy wizualizacji danych. Zasady tworzenia atrakcyjnych wykresów*, Helion, Gliwice 2015, str. 16. [↑](#footnote-ref-14)
15. C. O. Wilke, *Podstawy wizualizacji danych. Zasady tworzenia atrakcyjnych wykresów*, Helion, Gliwice 2015, str. 28. [↑](#footnote-ref-15)
16. Tamże. [↑](#footnote-ref-16)
17. J. Steele, N. Iliinsky, *Beautiful Visualization: Looking at Data through the Eyes of Experts (Theory in Practice)*, O'Reilly, 2010, str. 42. [↑](#footnote-ref-17)
18. J. Steele, N. Iliinsky, *Beautiful Visualization: Looking at Data through the Eyes of Experts (Theory in Practice)*, O'Reilly, 2010, str. 28. [↑](#footnote-ref-18)
19. C. Nussbaumer Knafic, *Storytelling with Data*, John Wiley & Sons, 2015, str. 62. [↑](#footnote-ref-19)
20. The Best Data Visualization Tools Of 2023, <https://www.forbes.com/advisor/business/software/best-data-visualization-tools/>, [07-04-2023]. [↑](#footnote-ref-20)
21. Ranking portali sprzedających nieruchomości, <https://ceo.com.pl/ranking-portali-sprzedajacych-nieruchomosci-53865#OTODOM>, [02-05-2023]. [↑](#footnote-ref-21)
22. Tamże. [↑](#footnote-ref-22)
23. Tamże. [↑](#footnote-ref-23)
24. *Best Programming Languages for Effective Web Scraping*, <https://oxylabs.io/blog/best-web-scraping-language>, [05-05-2023]. [↑](#footnote-ref-24)
25. *Power BI Vs Tableau: Difference and Comparison*, <https://www.simplilearn.com/tutorials/power-bi-tutorial/power-bi-vs-tableau#features_of_power_bi>, [14-05-2023]. [↑](#footnote-ref-25)