

Studium Magisterskie

Kierunek: Analiza danych

Specjalność: Big Data

Oskar Sękowski

Nr albumu 116226

**Wykorzystanie web scrapingu oraz algorytmów Machine Learning w przetwarzaniu i analizie danych z rynku nieruchomości**

Praca magisterska

napisana w Katedrze Analiz Ekonomicznych

pod kierunkiem naukowym

dr Sebastiana Zająca

Warszawa 2024

**Spis treści**

[1. Wprowadzenie 4](#_Toc156739016)

[1.1 Rynek nieruchomości w kontekście Big Data 5](#_Toc156739017)

[1.2 Web Scraping i uczenie maszynowe 6](#_Toc156739018)

[2. Przegląd literatury 13](#_Toc156739019)

[3. Materiały i metody badań 20](#_Toc156739020)

[3.1. Proces gromadzenia danych – web scraping 20](#_Toc156739021)

[3.2. Opis wykorzystanych modeli 23](#_Toc156739022)

[3.3. Miary oceny modeli 28](#_Toc156739023)

[4. Eksploracyjna analiza danych 31](#_Toc156739024)

[5. Wyniki modelowania 40](#_Toc156739025)

[5.1. Regresja liniowa 40](#_Toc156739026)

[5.2. Regresja wielomianowa 41](#_Toc156739027)

[5.3. Regresja wieloraka i LASSO 43](#_Toc156739028)

[5.4. Regresja ze zmiennymi kategorialnymi 44](#_Toc156739029)

[5.5. Drzewa decyzyjne i lasy losowe - regresja 47](#_Toc156739030)

[5.6. Algorytm k-NN 50](#_Toc156739031)

[5.7. Regresja logistyczna 52](#_Toc156739032)

[5.8. Drzewa decyzyjne i lasy losowe - klasyfikacja 53](#_Toc156739033)

[6. Podsumowanie 56](#_Toc156739034)

[Spis literatury 59](#_Toc156739035)

[Spis rycin 64](#_Toc156739036)

[Spis tabel 65](#_Toc156739037)

[Załączniki 66](#_Toc156739038)

[Streszczenie 75](#_Toc156739039)

# 1. Wprowadzenie

Rozwój technologii obserwowany we współczesnym świecie spowodował powstanie nowego, niewyczerpywalnego zasobu jakim są dane. Zasób ten przedstawiany jest w formie ustrukturyzowanej (dane tabelaryczne) lub nieustrukturyzowanej (dźwięk, obraz, tekst) (Zając, 2022). Rola danych jest tak istotna, że powszechnie uważane są za kluczowy element współczesnej ekonomii. Ogrom tego zasobu chyba najlepiej oddaje sformułowanie „big data”, które powstało specjalnie po to, by spróbować opisać ogromne bazy danych, procedury, ale   
i rozwijać wiedzę na temat tego, jak korzystać z zasobu danych (Winson-Geideman, Krause, 2016). Big Data wydaje się być szczególnie powiązana z rynkiem nieruchomości i to w wielu aspektach. Są one bowiem nabywane nie tylko w celach zaspokojenia potrzeb mieszkaniowych, ale i pod kątem inwestycyjnym. Informacje na ich temat są również gromadzone przez odpowiednie instytucje naukowe, urzędowe czy podmioty biznesowe w celu lepszego zrozumienia procesów zachodzących na tym rynku (Deberlanger, 2020). Głównymi aspektami w kontekście Big Data i rynku nieruchomości stały się w ostatnich latach trzy zagadnienia. Pierwszym z nich jest predykcja cen, która jest nie tylko wyzwaniem dla matematyków, statystyków, ale i pożądaną wiedzą biznesową (deweloperzy, pośrednicy). Do tego monitorowanie procesów zachodzących na rynku nieruchomości jest ważnym zagadnieniem makroekonomicznym. Drugim aspektem może być uczenie maszynowe, które dostarcza obecnie wiele narzędzi pozwalających na dokładną predykcję cen i analizę danych. Ostatnim jest natomiast pozyskanie dużej ilości odpowiednich danych, które stanowią integralną część uczenia maszynowego.

Nie bez znaczenia dla ostatniego zagadnienia pozostaje web scraping, dzięki któremu można pozyskać wielkie ilości danych zazwyczaj ze stron internetowych (Tsonev i in., 2022). Jak widać wszystkie trzy aspekty się wzajemnie uzupełniają i mogą udoskonalać.   
W najprostszym rozumieniu tego zjawiska dokładna predykcja cen oraz analiza rynku nieruchomości opiera się na wykorzystaniu metod uczenia maszynowego. Te z kolei mogą być zasilane dużą ilością danych dzięki web scrapingowi. Niniejsza praca jest próbą połączenia obu tych elementów oraz wypełnienie luki poznawczej w zakresie polskiej literatury na ten temat.

Cel pracy można podzielić na dwa aspekty. Pierwszym jest zbudowanie narzędzia web scraping oraz jego wykorzystanie do zbudowania zbioru danych. Drugim aspektem było przetestowanie wybranych metod uczenia maszynowego na wspomnianych danych w zakresie predykcji cen mieszkań oraz analizy danych związanych z rynkiem mieszkaniowym.

## 1.1 Rynek nieruchomości w kontekście Big Data

Jak wspomniano, w obecnych czasach dane są stałym i niezbędnym elementem wielu dziedzin gospodarki. Wykorzystywane przez przedstawicieli różnych specjalności przeniknęły zarówno do sfery osobistej, społecznej jak i zawodowej. Powszechność wykorzystania jak   
i ogromna ilość wykreowały pojęcie i techniczne aspekty big data. Dzięki temu możliwe stało się przedstawienie ogromnych, wzajemnie powiązanych baz danych i procedur stosowanych   
w celu uzyskania stosownej wiedzy. Big Data opisuje ogromne ilości danych, które mogą być strukturalne, semi-strukturalne oraz sub-strukturalne i mogą być one wykorzystywane w celu uzyskania określonych informacji (Sagiroglu, Sinanc, 2013).

Analiza wielkich zbiorów danych może przynieść wiele nowych kierunków badań dotyczących rynków nieruchomości. Jednym z przykładów może być rozwój chińskiego rynku nieruchomości, który dokonany został właśnie przy wsparciu big data. Wykorzystanie dużych zbiorów danych pomogło firmom związanym z nieruchomościami na zróżnicowanie inwestycji. Dane osobowe oraz te dotyczące rynku mogą stworzyć nowe możliwości dla firm zajmujących się obrotem nieruchomościami. Poza przykładowo analizą inwestycji big data na rynku nieruchomości znajduje głównie zastosowanie w prognozowaniu ich cen w określonym obszarze. Taka analiza możliwa jest dzięki przetwarzaniu dużych zbiorów danych na temat cen i znajdywaniu trendów, popytu na nieruchomości czy też analizie otoczenia danego lokalu. Czołowi autorzy badań zajmujący zagadnieniami big data na rynku nieruchomości wymieniają też kilka innych obszarów, w których może mieć ono duże znaczenie. Pierwszym z nich jest cyfryzacja rekordów, co oznacza przekształcenie dokumentów z formy papierowej. Może mieć to duże znaczenie dla firm obecnych na rynku nieruchomości, głównie ze względu na optymalizację pracy. Wszelkie niezbędne raporty, wyceny nieruchomości, pisma mogą być przechowywane w formie cyfrowej. Kolejnym obszarem big data w nieruchomościach mogą być informacje sensorowe pochodzące ze środowiska otaczającego np. budynki mieszkalne. Mowa tutaj o danych rejestrujących np. poziom hałasu, zanieczyszczenia powietrza, ale i też aktywności innych ludzi na danym obszarze. Informacje takie mogą być bardzo przydatne do wyceny danej nieruchomości zlokalizowanej w konkretnej okolicy (Kok i in., 2017).

Jak wskazują kolejni autorzy rynek nieruchomości od zawsze związany był z danymi począwszy od cen najmu, czy transakcyjnych dotyczących sprzedaży poprzez stopy procentowe, a kończąc na opłacalności inwestycji. W związku z tym dane w obszarze rynku nieruchomości można podzielić na trzy podstawowe obszary – finansowy, transakcyjny oraz fizyczny. Pierwsze z nich koncentrują się na informacjach dotyczących akcji firm działających na rynku nieruchomości. Obszar transakcyjny obejmuje natomiast dane na temat transakcji nieruchomości, kredytów hipotecznych, dzierżaw, podatków i sprawozdań finansowych. Ostatni z obszarów, czyli fizyczny dotyczy z kolei informacji na temat stanu nieruchomości, jej charakterystyki strukturalnej czy samej lokalizacji. Wymienione trzy typy danych są uznawane za „trzon” informacji pochodzących z rynku nieruchomości. Od lat 90-tych można mówić również o kolejnym obszarze danych w związku z nieruchomościami. Było to połączone   
z wprowadzeniem i rozprzestrzenianiem się Systemów Informacji Geograficznej (GIS), które umożliwiły zbieranie oraz analizę dodatkowych danych związanych z przestrzenią geograficzną. Mowa tutaj o zjawiskach przestrzennych poza konkretnymi granicami danej nieruchomości. Przykładami takich danych są np. informacje o sąsiedztwie, natężeniu ruchu, odległości od udogodnień, punktów usługowych (Oluwunmi i in., 2019).

Dane oraz rynek nieruchomości są więc ściśle powiązane i to zarówno w aspekcie ekonomicznym jak i akademickim, naukowym. Jednym z najbardziej dynamicznie rozwijających się tematów w tym zakresie jest również uczenie maszynowe, które wykorzystywane może być do predykcji cen nieruchomości, czyli jednej z najważniejszych cech (Varma i in., 2018). Wykorzystanie uczenia maszynowego w tym zakresie rozwinęło się szczególnie w ostatnich latach wraz z rozwojem Big Data. Wynika to z tego, że modele te bazując na cechach nieruchomości, ich wartości z poprzednich lat potrafią dokładniej przewidzieć ceny niż tradycyjne metody wykorzystywane w tym celu (Truong i in., 2020).   
W wielu przypadkach podobne analizy nie mogłyby się jednak odbyć gdyby nie web scraping, który jest niezwykle cennym narzędziem zarówno dla naukowców jak i osób zawodowo powiązanych z rynkiem nieruchomości. Wydaje się wręcz, że web scraping jest idealnym narzędziem dla uczenia maszynowego, ponieważ może dostarczać do tego obszaru bardzo cenny zasób, czyli dane, na których modele mogą być trenowane (Tsonev i in., 2022).

## 1.2 Web Scraping i uczenie maszynowe

Pierwsze z głównych pojęć wykorzystywanych w pracy, czyli web scraping można   
w skrócie zdefiniować jako narzędzie, konstrukcję do automatycznego pobierania, analizowania i organizowania danych z sieci. Pojęcie to znane jest również w literaturze anglojęzycznej jako „web data extraction”, „web harvesting” lub „web data mining”. Ogromną zaletą tego rozwiązania jest znaczne zautomatyzowanie czynności, dzięki czemu analiza danych jest przede wszystkim szybsza oraz bardziej dokładna niż gdyby była wykonywana ręcznie. W najprostszym znaczeniu web scraping to proces otrzymywania, zbierania interesujących użytkownika tekstowych informacji ze stron internetowych (Uzun, Novel, 2020). Celem tego procesu jest pobranie, wydobycie i przetworzenie pożądanych danych   
w dane strukturalne na potrzeby użytkownika (Patel, 2020). Web scraping jako metoda znajduje zastosowanie w wielu dziedzinach tj. np. życie społeczne, turystyka, e-commerce, Data Science, AI, cyberbezpieczeństwo, czy właśnie uczenie maszynowe. W przypadku tego ostatniego obszaru web scraping dostarcza dane, dzięki którym modele mogą uczyć się   
i udoskonalać swoje działanie. Te dwa narzędzia coraz częściej wykorzystywane są również do budowy specjalnych platform, na których prezentowane są dane w postaci wykresów, prognoz, rekomendacji, a wszystko to bazuje na algorytmach uczenia maszynowego zasilanych web scrapingiem niekiedy w czasie rzeczywistym (Tsonev i in., 2022). Choć wydaje się, że ta technika pozyskiwania danych jest bardzo dobrze rozwinięta, to wciąż wymaga doskonalenia, ponieważ obejmuje wiele różnych dziedzin tj. przetwarzanie tekstu, rozumienie semantyczne, sztucznej inteligencji oraz interakcji człowiek-komputer. Na tej podstawie można wyróżnić także kilka technik web scrapingu. Najstarsza z nich wymieniana w literaturze (Deberlanger, 2020) jest techniką opartą na kopiowaniu i wklejaniu przez człowieka. Znacznie bardziej wydajna bazuje na dopasowaniu wyrażeń regularnych dostępnych w językach programowania, czyli sekwencji znaków definiujących określony wzór wyszukiwania. Prostym przykładem może być tutaj sprawdzanie tagów na danej stronie internetowej. Ostatnia, prawdopodobnie najbardziej rozpowszechniona metoda web scrapingu to HTML parsing. W większości przypadków polega ona na pobraniu całego kodu źródłowego strony, a następnie wyodrębnieniu z niego interesujących informacji. Przykładowo sposób ten może obejmować tytuł strony, akapity, nagłówki, linki. Proces ten odbywa się przeważnie przy wykorzystaniu żądań HTTP do określonych adresów URL. Dzięki temu można uzyskać dostęp do zawartości strony internetowej – kodu źródłowego zazwyczaj w postaci HTML lub JSON.

W literaturze wymienia się również poszczególne etapy prawidłowego web scrapingu. Pierwszym z nich jest poznanie kodu źródłowego strony, czyli sprawdzenie w jakim języku powstała strona (np. HTML) i znalezienie znaczników zawierających interesujące informacje. Drugi etap stanowi napisanie kodu do pobierania danych. W tym przypadku z pomocą przychodzą różne języki programowania, aplikacje, czy biblioteki np. BeautifulSoup w Python. Ostatni etap polega na pobraniu, a następnie odpowiednim przetworzeniu pozyskanych danych (Zych, 2021).

Drugie pojęcie związane z pracą, czyli uczenie maszynowe (ang. *machine learning*) to dziedzina, której początki datuje się na lata 50-te XX wieku. Jej rozwój jest związany   
z amerykańskim informatykiem Arhurem Samuelem uważającym, że nauczenie grania komputera może stanowić początek do rozwiązywania bardziej skomplikowanych problemów (Bartłomiejczyk i in., 2020). Wspomnianemu autorowi przypisuje się również sformułowanie pierwszej definicji związanej z uczeniem maszynowym. Samuel (1959) po raz pierwszy określił *machine learning* jako zdolność komputerów do uczenia formułując następującą definicję: „Uczenie maszynowe to dziedzina nauki dająca komputerom możliwość uczenia się bez konieczności ich jawnego programowania”.

W późniejszych latach pojawiła się kolejna definicja związana z procesem uczenia się systemów. Simon (1983) stwierdził, że uczenie się oznacza zmiany w systemie mające charakter adaptacyjny, czyli taki, który pozwala systemowi następnym razem wykonać takie samo zadanie, ale w sposób bardziej efektywny. Wśród osób, które miały istotny wpływ na rozwój uczenia maszynowego wymienia się Breimana z Uniwersytetu Kalifornijskiego oraz Friedmana z Uniwersytetu Stanforda. W latach 80-tych XX wieku prowadzili badania nad rozwojem modelu drzew decyzyjnych (Bruce i in., 2021).

Wpływ na rozwój teoretycznych podstaw uczenia maszynowego miał również Michie (1991). Wskazał on, że system uczący się korzysta z zewnętrznych danych empirycznych   
w celu stworzenia i aktualizacji podstaw do udoskonalonego działania na zbliżonych danych   
w przyszłości oraz wyrażania tych podstaw w zrozumiałej i symbolicznej postaci.

W późniejszych latach pojawiła się również bardziej techniczna definicja dotycząca uczenia maszynowego: „Program komputerowy uczy się na podstawie doświadczenia   
E w odniesieniu do jakiegoś zadania T i pewnej miary wydajności P, jeśli jego wydajność (mierzona przez P) wobec zadania T wzrasta wraz z nabywaniem doświadczenia (Mitchell, 1997).

Bardziej współczesne pojęcie uczenia maszynowego można znaleźć przykładowo   
w pracy Flach’a (2012), który stwierdził, że *machine learning* jest nauką na temat algorytmów i systemów doskonalących swoją wiedzę oraz wyniki poprzez zdobywane doświadczenie.

Podobne postrzeganie uczenia maszynowego można wskazać wśród polskich autorów. Przykładowo Cichosz (2003) nazywa uczeniem maszynowym „każdą autonomiczną zmianę   
w systemie zachodzącą na podstawie doświadczeń, która prowadzi do poprawy jakości jego działania”.

Jak można zauważyć, w bardziej współczesnych opracowaniach dotyczących uczenia maszynowego m.in. (Michalski, Carbonell i in., 1983), (Weiss, Kulikowski, 1991), (Mitchell, 1997), (Bolc, Zaremba, 1992) pojawiają się wspólne elementy dla określania czym jest *machine learning*. Przede wszystkim wskazuje się, że jest on gałęzią sztucznej inteligencji, która zajmuje się automatycznym pozyskiwaniem wiedzy z danych w celu wykorzystania jej w przyszłości. Jest to również dziedzina nauki o programach i systemach mających podejmować określone działania, do realizacji których nie były wcześniej zaprogramowane. Wynika to z tego, że   
w całym procesie nie jest stosowane klasyczne programowanie. Istotą uczenia maszynowego jest to, że programy uczą się same z uzyskanych informacji na podstawie, których tworzone są wzorce i wykonywane określone działania tj. odnalezienie zależności oraz prawidłowości   
w danych. Stąd kluczowe znaczenie mają w tym przypadku analityka i eksploracja danych. Jak widać *machine learning* jest więc nauką interdyscyplinarną, która korzysta z podwalin matematyki, informatyki, robotyki czy statystyki. Celem tej dziedziny jest zastosowanie osiągnięć sztucznej inteligencji w tworzeniu systemu, który uczy się wykorzystując samodzielnie gromadzone doświadczenie i na tej podstawie nabywanie nowej wiedzy (Lasota-Kapczuk, 2021). Obecnie *machine learning* jest więc uznawany za dziedzinę sztucznej inteligencji wykazującą empiryczne „uczenie się” kojarzone z ludzką inteligencją, przy jednoczesnej zdolności do nauki oraz doskonalenia analizy z wykorzystaniem algorytmów obliczeniowych. Z praktycznego punktu widzenia jest on w dzisiejszych czasach bardzo często wykorzystywany do budowy modeli predykcyjnych. Nauka na podstawie danych pozwala tak dostosować algorytmy uczące by zwiększyć jakość prognoz (Korczak, 2019). Jednak obszary, w których *machine learning* może znaleźć zastosowanie są trudne do zliczenia. Choć początkowo głównym wykorzystaniem tych metod był filtr spamu rozwijany w latach 90-tych to od tego momentu nastąpił znaczący wzrost zastosowania uczenia maszynowego, który stanowi bazę dla różnego rodzaju aplikacji i usług np. tłumaczy, wyszukiwarek obrazów, rekomendacji produktów, wskazówek głosowych (Geron, 2023). Zastosowanie uczenia maszynowego ma prowadzić w teorii do tworzenia nowych pojęć, wykrywania nieodkrytych reguł, zależności między danymi, konstruowania zasad decyzyjnych, przyswajania przez maszyny pojęć i struktur opierając się na analogiach i uogólnieniach, zdobywaniu wiedzy   
i formułowaniu wniosków zrozumiałych dla człowieka. Praktyczne zastosowanie uczenia maszynowego jest więc bardzo szerokie. Może być wykorzystywane do analizy, poszukiwania zależności w obszernych bazach danych (big data), filtrowaniu informacji (np. wyszukiwarki internetowe), prognozowania w różnych gałęziach biznesu. Bazujące na historycznych danych modele uczenia maszynowego pozwalają m.in. pokazywać przyszłe zachowania klientów, wykrywać próby włamania, oszustwa, stwierdzać prawdopodobieństwo awarii czy wykrywać podejrzane transakcje (Lasota-Kapczuk, 2021).

Oprócz historycznych podstaw uczenia maszynowego należy również wspomnieć   
o jego technicznych aspektach. Obecnie można mówić o wielu jego typach, a podstawowego podziału można dokonać ze względu na (Geron 2023):

* sposób nadzorowania w fazie uczenia – nadzorowane, nienadzorowane, półnadzorowane, samonadzorowane
* możliwość uczenia się w czasie rzeczywistym – przyrostowe i wsadowe
* sposób pracy – przykładowo porównywanie punktów danych ze znanymi punktami

Pierwszy z typów wydaje się być obecnie najpowszechniej stosowanym podziałem. Ogólnym podział uczenia maszynowego ze względu na typ nadzorowania przedstawia poniższa rycina (rys. 1). Najprostsze podejście wyróżnia uczenie nadzorowane oraz nienadzorowane.   
W pierwszym podejściu model uczony jest na podstawie zbioru danych, gdzie dane wejściowe wykorzystywane są do wykrycia zależności. Na ogół danym tym towarzyszą etykiety albo klasy. W ten sposób na przekazanych algorytmowi danych zawierających etykiety może on poprawie nauczyć się klasyfikować, rozwiązywać dany problem. Typowym przykładem uczenia z nadzorem jest przewidywanie liczbowej wartości docelowej (regresja). Algorytm jest w stanie przewidzieć takie wartości po otrzymaniu wielu przykładów ich cech oraz wartości docelowych (Geron, 2023).

Uczenie maszynowe

Nadzorowane

Nienadzorowane

Klasyfikacja

Klasteryzacja

Regresja

Sieci neuronowe

Lasy losowe

k-NN

Regresja liniowa

Drzewa decyzyjne

Metody Ensemble

SVR

Naiwny klasyfikator bayesowski

Sieci neuronowe

k-NN

Analiza dyskryminacyjna

SVM

K-Means

Analiza hierarchiczna

Gaussian Mixture

Sieci neuronowe

Ukryte modele Markov Markowa

Rysunek 1. Schemat uczenia maszynowego

*Źródło:* Opracowanie własne na podstawie Ja’afar i in., 2021

Jak widać na rycinie uczenie nadzorowane można podzielić na dwa główne zagadnienia – regresyjne oraz klasyfikacyjne. Do algorytmów z pierwszej grupy należą sieci neuronowe, lasy losowe, k-NN, regresja liniowa, drzewa decyzyjne, SVM, metody ensembe (bagging, boosting, stacking, voting). Niektóre z tych metod wykorzystywane są również do zagadnień klasyfikacyjnych – sieci neuronowe, k-NN, czy SVM. Poza nimi wyróżnia się jeszcze naiwny klasyfikator bayesowski i analizę dyskryminacyjną (Ja’afar i in., 2021).

W odróżnieniu od uczenia z nauczycielem (nadzorowane) techniki nienadzorowane wykorzystują dane uczące nieoznakowane. Innymi słowy system próbuje znaleźć zależności między danymi, ale bez ingerencji człowieka (Geron, 2023). W tym przypadku głównym zagadnieniem jest klasteryzacja, w której można wyróżnić następujące algorytmy uczenia maszynowego: K-Means, K-meddis, Fuzzy C-Means, Hierarchical, Gaussian Mixture, Sieci neuronowe, Hidden Markov Model (Ja’afar i in., 2021).

W przypadku uczenia nienadzorowanego typowym zagadnieniem może być analiza skupień, którą można wykorzystać do odnalezienia grup podobnych użytkowników. Innym przykładem jest tutaj redukcja wymiarowości, której zadaniem jest uproszczenie danych bez zbyt dużej utraty informacji (np. proces wydobywania cech). Bardzo ważnym zagadnieniem jest także wykrywanie anomalii pozwalające wytypować obserwacje nietypowe (Geron, 2023).

Połączeniem tych dwóch technik jest uczenie półnadzorowane, które najczęściej stanowi kombinację algorytmów nadzorowanych i nienadzorowanych. Ta część uczenia maszynowego występuje wtedy, gdy mamy do czynienia z większością nieoznakowanych   
i jakąś częścią oznakowanych informacji w zbiorze. Całkowicie nieoznakowane zestawy danych są natomiast wykorzystywane w przypadku uczenia samonadzorowanego, którego celem jest wygenerowanie w pełni oznakowanego zbioru. Ostatni typ związany   
z nadzorowaniem w fazie uczenia to uczenie przez wzmacnianie, które odbiega znacznie od wcześniej opisanych technik. W tym procesie system uczący jest tzw. Agentem, który może obserwować środowisko, dobierać, wykonywać różne czynności oraz odbierać za to nagrody lub kary. W ten sposób samodzielnie uczy się najlepszej strategii działania tak aby otrzymać jak największą nagrodę (Geron, 2023).

# 2. Przegląd literatury

Rynek nieruchomości oraz metody uczenia maszynowego były przedmiotem licznych publikacji naukowych, głównie zagranicznych. Główne zagadnienie poruszane w pracach   
o wspomnianej tematyce dotyczy predykcji cen nieruchomości, dlatego też poniższy przegląd literatury został oparty w przeważającej części na tym zagadnieniu.

Jednym z pierwszych badań w obszarze *machine learning* oraz nieruchomości były badania przeprowadzone przez Park i Bae (2014). Celem opracowania było rozwiązanie problemu klasyfikacji za pomocą modelu lasów losowych oraz wykorzystanie regresji wykorzystując klasyfikator Bayesowski. Dane dotyczyły 5359 nieruchomości w miejscowości Fairfax, w stanie Virginia i pochodziły z regionalnego systemu informacyjnego. Najdokładniejszym algorytmem prognozującym ceny mieszkań okazał się być model RIPPER.

Z tego samego roku pochodzi również projekt badawczy dotyczący uczenia maszynowego i predykcji cen nieruchomości w Montrealu (Pow i in., 2014). Autorzy publikacji wykorzystali dane pochodzące z jednego z kanadyjskich serwisów ogłoszeniowych. Na podstawie zmiennych tj. lokalizacja geograficzna, powierzchnia użytkowa, liczba pokoi oraz zmiennych dodatkowych tj. odległość od najbliższego posterunku policji, straży pożarnej oszacowali model dotyczący przewidywania cen ofertowych jak i transakcyjnych. W tym celu wykorzystali metody regresji liniowej, SVR, k-NN, drzew decyzyjnych oraz lasy losowe. Jeżeli chodzi o predykcje cen ofertowych najlepszymi rezultatami charakteryzowały się algorytmy k-NN oraz lasów losowych z błędem na poziomie 0,09. Dla ceny transakcyjnej najmniejszy błąd (0,02) wykazywał model lasów losowych.

Również w 2014 roku ukazała się publikacja, której autorami byli Mu i in. (2014). Celem pracy było przetestowanie wybranych algorytmów uczenia maszynowego do przewidywania cen domów w dzielnicach podmiejskich Bostonu. Wśród metod wykorzystanych w artykule znalazły się modele SVM, LSSVM oraz PLS. Jak wynika z badania dwa pierwsze algorytmy wykazywały lepsze wyniki predykcyjne w przypadku danych nieliniowych, a PLS natomiast dla zmiennych o związkach liniowych.

Algorytmy uczenia maszynowego były również wykorzystywane do przewidywania cen gruntów. Przykładem takich badań może być praca Samapathkumar’a i in. (2015), którzy przeprowadzili swoje analizy dla jednego ze stanów Indii z wykorzystaniem regresji wielorakiej oraz sieci neuronowych. Modele te były trenowane na podstawie średnich miesięcznych cen z lat 1997-2011, a następnie walidowane na danych z roku 2012 i 2013. Dla dwóch następnych okresów wartość gruntów została oszacowana za pomocą wspomnianych modeli. Wyniki wskazały na większą dokładność (*accuracy*) sieci neuronowych.

Szczegółową analizę możliwości wykorzystania uczenia maszynowego w predykcji cen nieruchomości wykonali również Yu i Wu (2016). Celem pracy było opracowanie modelu regresyjnego oraz klasyfikacyjnego do oszacowania cen nieruchomości na podstawie konkretnych zmiennych. W swojej pracy autorzy wykorzystali techniki regresyjne - LASSO, regresja grzbietowa, SVM, lasy losowe. Przewidywania przedziałów wartości zostały natomiast wykonane w oparciu o klasyfikator bayesowski, regresję logistyczną, klasyfikator SVM oraz klasyfikator lasów losowych. Dla problemu klasyfikacji najlepszym modelem okazał się być model SVC, który charakteryzował się dokładnością (*accuracy)* na poziomie 67,00%. Uwzględniając analizę PCA parametr ten wzrastał do ponad 69,00%. Z kolei problem regresyjny najlepiej został rozwiązany przez model SVR (Gaussian kernel). Błąd średniokwadratowy RMSE tego modelu wynosił prawie 53,00%. Wyniki badań wskazują również, że największe znaczenie w przewidywaniu cen nieruchomości miały zmienne tj. powierzchnia użytkowa, pokrycie dachu oraz otoczenie.

Kolejnym przykładem badań w zakresie tematu pracy może być analiza przeprowadzona przez Lu i in. (2017) przy okazji międzynarodowej konferencji. Jak wspominają autorzy bardzo popularne indeksy cen nieruchomości odnoszą się ogólnie do sytuacji na rynku nieruchomości. Problematyczna staje się natomiast wycena jednego konkretnego obiektu, ponieważ zależy od wielu cech tj. np. lokalizacja, typ, powierzchnia, rok budowy itd. Celem pracy było wykorzystanie połączonych regresji LASSO oraz Gradient Boosting do predykcji ceny konkretnej, indywidualnej nieruchomości. Jak wynika z badania kombinacja tych dwóch metod przyniosła najbardziej pożądane rezultaty i najlepsze oceny modelu.

Ceny nieruchomości mogą być także przedmiotem klasyfikacji. Przedmiotem badania Banerjee i Dutta (2017) były zmiany cen domów i wykorzystanie algorytmów uczenia maszynowego tj. SVM, lasy losowe oraz ANN do przewidywania, czy wartość danej nieruchomości spadnie czy wzrośnie. Wyniki badań opierają się na zmiennej dyskretnej, którą reprezentują dwie wartości 0 (spadek ceny) lub 1 (wzrost ceny) klasyfikującej daną nieruchomość. Największą dokładnością (*accuracy*) oraz precyzją (*precision*) charakteryzował się model lasów losowych – odpowiednio 86,00% oraz 79,00%. W przypadku pozostałych miar oceny modelu najlepsze wyniki uzyskały modele SVM (*sensitivity)* i ANN (*specifity*). Jak wskazują autorzy, pomimo, że metoda lasów losowych charakteryzowała się najlepszymi wynikami, to wykazywała podatność na nadmierne dopasowanie.

W 2017 roku ukazała się również praca na temat prognozowania cen nieruchomości   
z wykorzystaniem modeli regresji wielorakiej (Manjula i in., 2017). Modele te zostały zbudowane w oparciu o zbiór danych liczący 21 tys. domów w regionie King County w Seattle. Zbiór ten został podzielony na dane treningowe oraz testowe w proporcji 80:20. W analizie wykorzystano następujące informacje na temat nieruchomości: powierzchnia użytkowa, cena, data sprzedaży, liczba pokoi, piętro, czy front domu jest położony od strony zbiornika wodnego, rok budowy, kod pocztowy miejsca, długość oraz szerokość geograficzna. Każdy model zawierający różne zestawy zmiennych i stopnie złożoności został oceniony na podstawie błędu średniokwadratowego RMSE. Wyniki pracy wskazywały na duże niedopasowanie modelu liniowego. Z kolei modele złożone wykazywały cechy przeuczenia.

Regresja liniowa jako jeden z algorytmów uczenia maszynowego został wykorzystany w badaniach Ghosalkar i Dhage (2018). Celem pracy było opracowanie narzędzia, które będzie przewidywać ceny nieruchomości w Bombaju dla klientów, którzy dzięki temu mogliby określać swoje finansowe plany i potrzeby związane z kupnem nieruchomości. Oszacowany model regresji liniowej cechował się błędem na poziomie 0,37.

Modele uczenia maszynowego były również przedmiotem analizy porównawczej   
w stosunku do modelu hedonicznego cen nieruchomości. W swojej pracy Ceh i in. (2018) wykorzystali zbiór danych zawierający 7,4 tys. ofert zebranych w okresie od 2008 do 203 roku w stolicy Słowenii – Lublanie. Analiza wykazała, że model lasów losowych cechował się lepszymi zdolnościami predykcyjnymi niż model hedoniczny.

Prognozowanie cen nieruchomości z wykorzystaniem uczenia maszynowego było także przedmiotem badań w Melbourne w Australii (Phan, 2018). Autor wykorzystał zbiór danych pochodzący z Kaggle zawierający prawie 39 tys. danych i 21 zmiennych. Wspomniana baza zawierała informacje na temat transakcji sprzedaży domów z okresu 2016-2018. Dodatkowo do każdej z nich przypisane zostały zmienne tj. cena, data sprzedaży, metoda sprzedaży, dane lokalizacyjne (adres, odległość do dzielnicy biznesowej miasta, kod pocztowy, długość   
i szerokość geograficzna) oraz dane związane z budynkiem tj. typ domu, liczba pokoi, liczba łazienek, powierzchnia działki. Predykcja cen nieruchomości przeprowadzona została   
z wykorzystaniem regresji liniowej, regresji wielomianowej, drzewa regresyjnego, sieci neuronowych oraz SVM w połączeniu z PCA i *stepwise selection*. Z pracy wynika, że drzewa regresyjne charakteryzowały się podobnymi właściwościami co regresja liniowa – zbliżone MSE na zbiorach testowym i walidacyjnym. Największe wartości MSE wystąpiły natomiast   
w przypadku sieci neuronowych (około 0,27).

Na podstawie danych z Kaggle swoje badania oparli również Fan i in. (2018). Wykorzystali oni zbiór danych z okresu 2006-2010, w którym znalazło się 37 zmiennych numerycznych oraz 43 kategorycznych. Ceny nieruchomości były prognozowane   
z wykorzystaniem kilku algorytmów uczenia maszynowego tj. regresja grzbietowa, regresja LASSO, lasy losowe, SVR, XGB.

Z metod uczenia nadzorowanego w celu przewidywania cen wynajmu skorzystali natomiast Hu i in. (2019). Celem publikacji była predykcja stawek najmu w chińskiej miejscowości Shenzhen. Spośród algorytmów lasów losowych, ETR, GBR, SVR, MLP-NN, k-NN najlepszymi parametrami charakteryzował się model lasów losowych.

Na temat cen wynajmu i uczenia maszynowego pisali także Neloy i in. (2019), których celem pracy było przewidywanie stawek na przykładzie Dhaki w Bangladeszu. W pracy wykorzystano kilka algorytmów tj. MLP-NN, RF, SVM, drzewa decyzyjne, regresja LASSO, regresja grzbietowa. Do oceny modeli wykorzystano błąd MSE, który najmniejszą wartość osiągnął w przypadku modelu lasów losowych.

Podobne wnioski wynikają z publikacji, której autorem był Chatzidis (2019). Analiza polegała na wykorzystaniu modeli LGBM, XGBM, CatBoost oraz lasów losowych w celu predykcji cen domów w Holandii. W tym przypadku najlepszym rozwiązaniem okazały się być modele CatBoost oraz RF z dokładnością (*accuracy*) na poziomie 90,00%.

Jednym z nowszych badań związanych z dziedziny uczenia maszynowego i cen nieruchomości są badania Kuvalekar i in. (2020). Celem pracy było przewidzenie rynkowej wartości nieruchomości w Bombaju. Dane pozyskane zostały z serwisów ogłoszeniowych   
i zawierały zmienne charakterystyczne dla nieruchomości tj. lokalizacja, powierzchnia użytkowa, wiek nieruchomości, kod pocztowy. Dane zostały poddane modelowaniu   
z wykorzystaniem kilku algorytmów uczenia maszynowego – SVM, lasy losowe, regresja liniowa, regresja wieloraka, drzewa decyzyjne, k-NN. Najlepszym modelem predykcyjnym okazały się drzewa decyzyjne, które charakteryzowały się dokładnością (*accuracy)* na poziomie 89,00%.

W 2020 roku ukazała się publikacja Troung i in., której celem było wykorzystanie metod uczenia maszynowego w predykcji cen nieruchomości w Pekinie. Praca oparta była na zbiorze liczącym ponad 300 tys. danych. W badaniu wykorzystano modele lasów losowych, XGBoost, LGBM, regresji LASSO i grzbietowej oraz SGR. Jak wynika z pracy algorytm lasów losowych cechował się najmniejszym błędem na zbiorze treningowym, ale wykazywał cechy przeuczenia. Najlepszą metodą okazała się modele LASSO i regresji grzbietowej ze względu na proces generalizacji. Z kolei SGR choć jak podkreślają autorzy cechuje się skomplikowaną architekturą, to jest najlepszym algorytmem pod kątem dokładności (*accuracy*).

Dane dotyczące indyjskich nieruchomości zostały wykorzystane w pracy Thamarai   
i Malarvizhi (2020). Na podstawie zbioru danych zawierającego informacje na temat liczby pokoi, roku budowy budynku, dostępności transportowej, bliskości szkół i sklepów zbudowane zostały modele drzewa decyzyjnego, lasów losowych oraz regresji wielorakiej. Ostatni z modeli charakteryzował się lepszą predykcyjnością cen niż drzewo regresyjne.

Napisana przez Hong’a (2020) praca dotyczyła porównania klasycznego modelu hedonicznego oraz modeli *machine learning* (XGBM, LGBM, CatBoost) do prognozowania cen transakcyjnych w stolicy Korei Południowej, czyli Seulu. Praca została zrealizowana na podstawie ponad 620 tys. obserwacji z okresu 2009-2019 roku. Jak wynika z pracy modele uczenia maszynowego charakteryzowały się większymi zdolnościami predykcyjnymi.   
W dodatku algorytm CatBoost wykazywał bardzo dobre zdolności predykcyjne nawet przy występowaniu wartości odstających.

Analiza porównawcza klasycznego modelu hedonicznego została również przeprowadzona w porównaniu do algorytmu lasów losowych (Hong i in., 2020). W tym celu wykorzystano informacje na temat 16,6 tys. mieszkań i ich cen z lat 2006-2017. Wspomniana metoda uczenia maszynowego wykazywała bezwzględnie lepsze parametry predykcyjne.

Jedną z nowszych prac związanych z uczeniem maszynowym i rynkiem nieruchomości były badania przeprowadzone przez Grybuskas i in. w 2021 roku. Celem pracy była prognoza cen nieruchomości i określenie spadku ich wartości podczas pandemii COVID-19. Autorzy wykorzystali prawie 19 tys. ofert sprzedaży nieruchomości w Wilnie. Ten etap badania został przeprowadzony w oparciu o algorytm wykorzystujący web scraping. Program ten został napisany w języku Python wykorzystując popularne biblioteki tj. BeautifulSoup oraz Selenium. Pozwoliło to na zebranie prawie 19 tys. ofert nieruchomości. Dla każdej z nich wyodrębniono 16 zmiennych – dzielnica miasta, adres, cena, liczba pokoi, powierzchnia, piętro, liczba pięter w budynku, miejsce na liście pod względem ceny, rok budowy budynku, odległość od sklepu, przedszkola, szkoły, typ budynku, rodzaj ogrzewania, dostępność oraz zmiany ceny. Autorzy   
w swojej pracy wykorzystali 15 algorytmów uczenia maszynowego tj. CatBoost, XGB, LGBM, lasy losowe, drzewa decyzyjne, GBC, analiza dyskryminacyjna, regresja logistyczna, regresja grzbietowa, naiwny klasyfikator bayesowski, ADA Boost, k-NN, SVM (kernel liniowy). Jak wynika z badań sektor nieruchomości jest sektorem dosyć odpornym na sytuacje pandemiczne, ponieważ zanotowane spadki cen nieruchomości nie były duże. W ciągu trwającego 4 miesiące badania ceny ofertowe sprzedaży nieruchomości spadły o około 7%, a najmu o 4%. Spośród wszystkich przeanalizowanych modeli najlepsze wyniki osiągnął algorytm XGB, ale różnice   
w miarach jakości pomiędzy poszczególnymi modelami były niewielkie.

W tym samym roku opublikowana została praca Khare’a i in. (2021), której celem była estymacja cen nieruchomości w oparciu o techniki *data mining*. Zbiór danych wykorzystany   
w pracy obejmował ponad 7,1 tys. rekordów oraz 19 zmiennych tj. np. powierzchnia nieruchomości, liczba sypialni, łazienek, pokoi, kondycja i typ budynku czy cechy lokalizacyjne. Zbiór danych został przeanalizowany z wykorzystaniem regresji liniowej, regresji wielomianowej, drzewa regresyjnego oraz lasów losowych. Do oceny zbudowanych modeli użyte zostały MSE, RMSE, MSLE oraz R2. Najlepszymi wynikami charakteryzował się model regresji wielomianowej z wykorzystaniem 18 zmiennych.

Jedną z najnowszych prac wykorzystujących web scraping oraz modele uczenia maszynowego jest praca Putri i in. (2023), w której dane zostały pobrane z publicznego serwisu ogłoszeniowego. Oferty te dotyczyły nieruchomości w jednej z indonezyjskich prowincji. Zebrane informacje na temat powierzchni domów, powierzchni działek, liczby pokoi oraz łazienek zostały poddane modelowaniu z wykorzystaniem 5 modeli regresyjnych tj. regresja wieloraka, LASSO, lasy losowe, regresja wielomianowa. Celem pracy była predykcja wartości. Ocena modeli została przeprowadzona z wykorzystaniem R2, RMSE oraz kroswalidacji. Dwie pierwsze miary pokazały, że najlepszymi modelami do predykcji cen są regresja wieloraka   
i regresja LASSO.

Za pomocą web scraping dane pozyskali również Khan i in. (2023). Zadaniem skonstruowanego scrapera było pobranie ogłoszeń nieruchomości w Karachai (Pakistan)   
z jednego z serwisów ogłoszeniowych. W ten sposób pobranych zostało ponad 15 tys. ogłoszeń, z których każde posiadło następujące zmienne: cena, lokalizacja, liczba pokoi, łazienek, powierzchnia, data, typ nieruchomości (domy oraz mieszkania). Dane te zostały następnie podzielone na zbiory treningowe i testowe oraz wykorzystane do budowy siedmiu modeli uczenia maszynowego tj. ADABoost, lasy losowe, Gradient Boosting, regresji grzbietowej i LASSO, sieci neuronowych. Najlepsze wyniki osiągnął model lasów losowych, który charakteryzował się najwyższymi wartościami R2 oraz najniższymi błędami MAE, MSE, RMSE.

Modele regresyjne były również przedmiotem publikacji Liu (2023), który wykorzystał zbiór danych pochodzący z Kaggle zawierający 1259 obserwacji i 81 zmiennych. Zmienną objaśnianą podobnie jak w poprzednich pracach była cena domów. Spośród modeli regresyjnych najlepszym okazała się być regresja LASSO z parametrem λ równym 0,001. Model ten osiągnął 90,10% dokładności (*accuracy*) i najmniejszy błąd. Jednak niewiele niższą dokładność uzyskały również regresja liniowa oraz grzbietowa.

# 3. Materiały i metody badań

W pracy wykorzystane zostały dane pochodzące z jednego z polskich serwisów ogłoszeniowych z branży nieruchomości. Portal www.gratka.pl należy do grupy Morizon-Gratka Sp. z o.o., która jest obecnie jednym z wiodących serwisów ogłoszeniowych w naszym kraju. Świadczyć może o tym fakt, że portale należące do grupy co miesiąc odwiedza około   
4 mln użytkowników. Gratka.pl obecna jest na polskim rynku od ponad 20 lat, a w bazie ogłoszeń można znaleźć nie tylko nieruchomości ale i ogłoszenia z kategorii praca, motoryzacja, sport, czy noclegi.

Obecnie na stronach serwisu znaleźć można ponad 160 000 ogłoszeń z branży nieruchomości. Większość z nich (87 000) dotyczy mieszkań. Pozostałe mniej liczne kategorie stanowią natomiast domy (36 000), lokale użytkowe (32 000), działki i grunty (44 000) oraz garaże, pokoje i inwestycje deweloperskie. Filtry zamieszczone na portalu pozwalają użytkownikom na wyszukiwanie ofert nieruchomości poprzez lokalizację, cenę, rodzaj ogłoszenia, kategoria rynku, powierzchnię.

Wszystkie poniżej opisane metody tj. web scraping, modele uczenia maszynowego, miary jakości modeli zostały zbudowane w języku Python z wykorzystaniem środowiska Jupyter Notebook. Fragmenty odpowiednich kodów z uwagi na objętość zamieszczone zostały w załączniku.

## 3.1. Proces gromadzenia danych – web scraping

Dane z serwisu pobrane zostały przy użyciu narzędzi skonstruowanych w języku Python. W przypadku web scrapingu wykorzystane zostały popularne biblioteki tj. *BeautifulSoup, requests* oraz *json*. Pierwsza z nich służy do analizowania dokumentów HTML i XML, dzięki czemu możliwe jest wyodrębnienie danych z tych języków. Biblioteka *requests* z kolei jest biblioteką, która służy do wykonywania zapytań HTTP. Umożliwia ona wysyłanie żądań HTTP oraz obsługę odpowiedzi. Ostatnia z nich, czyli biblioteka *json* jest narzędziem służącym do kodowania i dekodowania danych w formacie JSON. Biblioteki te pozwalają na komunikację z serwisem, pobranie kodu strony (HTML lub JSON), a następnie na ekstrakcję danych z kodów źródłowych. Procesy te zostały przeprowadzone w oparciu o zbudowaną klasę zawierającą następujące metody (funkcje): *get\_offers, get\_url, get\_location, get\_details, get\_area, get\_rooms, get\_level, get\_prices\_all, get\_main\_price, get\_sq\_price, get\_parse\_offer*.

Pierwsza z wspomnianych funkcji jako argument przyjmowała numer strony z serwisu internetowego (*page\_number*) i wysyłała żądanie HTTP oraz czeka na odpowiedź serwisu. Następnym krokiem jest pobranie zawartości danej strony internetowej, czyli kodu HTML, który później jest przetwarzany za pomocą biblioteki *BeautifulSoup*. W końcowym etapie zwracane są elementy kodu, które zawierają elementy *div* o klasie '*listing\_teaserWrapper*'. Efektem tej funkcji były więc zawartości kodów HTML poszczególnych stron serwisu, w której można znaleźć informacje dotyczące konkretnych ofert.

Kolejnym krokiem było napisanie funkcji, które miały pobierać wymagane informacje dla każdej oferty w kolejnych iteracjach. Informacje te można pozyskać znajdując odpowiednie fragmenty kodu HTML.

Funkcja *get\_url* jako argument przyjmowała zmienną *offer*, która reprezentowała daną ofertę nieruchomości. Jej zadaniem było wyszukanie zdefiniowanych elementów i pobranie adresu *url* iterowanej oferty oraz zwrócenie jej w postaci linku do ogłoszenia. Podobnym działaniem charakteryzowała się funkcja *get\_location*, która odnosiła się do informacji dotyczących lokalizacji ofert. Zadaniem tego fragmentu kodu było wyszukanie odpowiednich elementów strony zawierających miasto, dzielnicę oraz ulicę danej nieruchomości. Kolejną funkcją wykorzystaną w klasie do scrapowania danych była funkcja *get\_details* również przyjmująca argument *offer*. Jej zadaniem było zapisanie i zwrócenie informacji odnoszących się do powierzchni mieszkania, liczby pokoi oraz piętra. Z tak pozyskanego kodu należało jeszcze wyciąć wspomniane informacje. Posłużyły do tego funkcje *get\_area, get\_rooms* oraz *get\_level*, które jako argument przyjmowały efekt poprzednio opisanej funkcji. W pierwszym kroku za pomocą metody .*get\_text()* pochodzącej z biblioteki *BeautifulSoup* pobierana była zawartość tekstowa elementu HTML bez uwzględniania tagów. Tak oczyszczony fragment tekstu był następnie dzielony na postawie spacji i zapisywany w postaci listy. Ostatecznie odpowiedni element listy był przekształcany na liczbę zmiennoprzecinkową *float*.

Jedną z ważniejszych funkcji dla zrealizowania celu pracy była funkcja odpowiadająca za pobieranie i przetwarzanie informacji związanych z ceną mieszkań. W tym przypadku skonstruowane zostały trzy funkcje. Pierwsza z nich *get\_prices\_all* poza wyszukiwaniem   
i zapisywaniem odpowiedniego fragmentu HTML wyodrębniała z niego tekst, a następnie za pomocą metody .*replace()* zastępowała wszelkie niepożądane znaki tj. symbol waluty, zbędne spacje. W tym przypadku niezbędne było również umieszczenie w funkcji pętli *for*, której zadaniem było zwrócenie braku danych w przypadku gdy w serwisie nie widniała cena nieruchomości. Uzyskane w ten sposób wyniki zawierały jednak dwie wartości – cenę ogólną mieszkania oraz cenę w przeliczeniu na metr kwadratowy. Dwie pozostałe funkcje, czyli *get\_main\_price* oraz *get\_sq\_price* wyodrębniały odpowiednie wartości i zwracały je ostatecznie jako liczbę zmiennoprzecinkową.

Ostatnia funkcja o nazwie *parse\_offer* to funkcja przyjmująca jako argument fragment struktury kodu HTML (*offer*). Głównym zadaniem tej funkcji było wywoływanie wyżej opisanych poleceń. Ostatecznym wynikiem jej działania było zwrócenie zmiennych *url, location, area, rooms, level, prices, main\_price oraz sq\_price*.

Uruchomienie i działanie całej klasy opierało się o pętlę *for*. Przed jej uruchomieniem należało stworzyć pustą listę, do której będą dodawane informacje pochodzące z działania funkcji. Pętla jako argumenty przyjmowała zakres stron internetowych, a następnie dla tego każdej ze stron wywoływała odpowiednie funkcje pobierające kod HTML oraz wydobywając informacje dla każdej oferty mieszkania. W pętli zamieszczono również blok *except*, który miał obsługiwać wyjątki w przypadku niepowodzenia parsowania.

Drugie narzędzie napisane w ramach web scrapingu oparte było natomiast o dostępne na stronie skrypty JSON, które zawierały wszystkie niezbędne informacje o każdym ogłoszeniu tj. adres *url*, lokalizacja (miasto, dzielnica, adres), liczba pokoi, piętro, powierzchnia mieszkania, cena. Dzięki dostępności skryptów w tym formacie oraz skonstruowanemu narzędziu dodatkowo dla każdej oferty pobrano również informacje na temat długości   
i szerokości geograficznej oraz roku budowy budynku, w którym mieściło się mieszkanie. Narzędzie oparte o skrypty JSON składało się natomiast z dwóch funkcji – *get\_offers* oraz *proces\_page*. Działanie i budowa pierwszej z nich była analogiczna jak w przypadku narzędzia opartego o kod HTML z tą różnicą, że funkcja pobierała informacje z formatu JSON   
w przypadku gdy odnalazła element *<script>*. Druga funkcja jako argument (*json\_text*) przyjmowała wyodrębniony wcześniej fragment kodu i zapisywała go następnie do odpowiedniej zmiennej. W funkcji można odnaleźć również inicjalne utworzenie pustych list, z których każda odpowiadała pożądanej informacji na temat mieszkania. Kolejnym elementem funkcji była pętla *for*, która w zmiennej zawierającej tekst JSON odnajdywała odpowiednie elementy i na tej postawie wyróżniała informacje o ofertach. W kolejnych iteracjach dla każdej oferty wyodrębniane były wspomniane cechy mieszkań. Na tej podstawie tworzony był *data frame*, w którym dodatkowo zapisywana była informacja, czy mieszkanie pochodzi z rynku wtórnego czy pierwotnego. Następnie przy użyciu kolejnej pętli *for* wywoływane były wspomniane funkcje dla stron w odpowiednim zakresie.

Proces pobierania danych został rozpoczęty na początku stycznia 2024 roku i trwał   
3 dni. Ostatecznie działanie wyżej opisanego skryptu pozwoliło na pobranie 42 422 ogłoszeń mieszkań z 18 miast wojewódzkich w Polsce. Każda oferta posiadała 14 pobranych informacji, co w efekcie pozwoliło na utworzenie zbioru danych liczącego 593 908 danych. Dokładną liczbę zgromadzonych ogłoszeń wraz z podziałem na typ rynku przedstawiono w tabeli 1.

Tabela 1. Liczba pobranych i dostępnych ogłoszeń mieszkań

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Miasto** | **Pobrane ogłoszenia** | | | **Ogłoszenia dostępne** | | |
| **Wtórny** | **Pierwotny** | **Razem** | **Wtórny** | **Pierwotny** | **Razem** |
| Warszawa | 5 344 | 4 560 | 9 904 | 5 337 | 4 560 | 9 897 |
| Kraków | 2 944 | 2 920 | 5 864 | 2 953 | 2 924 | 5 877 |
| Gdańsk | 1 920 | 3 339 | 5 259 | 1 929 | 3 340 | 5 269 |
| Łódź | 1 472 | 2 412 | 3 884 | 1 490 | 2 411 | 3 901 |
| Białystok | 230 | 178 | 408 | 230 | 178 | 408 |
| Bydgoszcz | 1 044 | 1 091 | 2 135 | 1 043 | 1 091 | 2 134 |
| Lublin | 519 | 257 | 776 | 519 | 257 | 776 |
| Olsztyn | 450 | 186 | 636 | 451 | 186 | 637 |
| Toruń | 197 | 167 | 364 | 197 | 167 | 364 |
| Szczecin | 675 | 532 | 1 207 | 675 | 532 | 1 207 |
| Zielona Góra | 89 | 47 | 138 | 89 | 49 | 138 |
| Gorzów Wielkopolski | 108 | 4 | 113 | 107 | 5 | 113 |
| Poznań | 785 | 2 336 | 3 121 | 786 | 2 336 | 3 122 |
| Wrocław | 1 984 | 3 442 | 5 426 | 1 983 | 3 439 | 5 422 |
| Katowice | 507 | 1 128 | 1 635 | 505 | 1 127 | 1 632 |
| Rzeszów | 176 | 553 | 729 | 176 | 553 | 729 |
| Opole | 186 | 59 | 246 | 186 | 60 | 246 |
| Kielce | 307 | 270 | 577 | 307 | 270 | 577 |

*Źródło:* Opracowanie własne

Jak można zauważyć działanie skryptu przebiegło w prawidłowy sposób. Dla większości miasto pobrano wszystkie możliwe oferty, które były dostępne na portalu. Najwięcej ofert dotyczyło Warszawy (9 904), Krakowa (5 864), Gdańska (5 259) oraz Wrocławia (5 426). Najmniej dostępnych ogłoszeń lokali mieszkalnych w momencie przeprowadzania web scrapingu było natomiast w Gorzowie Wielkopolskim (113).

## 3.2. Opis wykorzystanych modeli

Jak wspomniano w wielu dziedzinach życia społecznego, zawodowego, czy naukowego można wykorzystywać uczenie maszynowe i stosować różne jego algorytmy.   
W odniesieniu do tematu pracy, czyli modelowania i prognozowania cen nieruchomości można jednak wyróżnić pewne prawidłowości w zakresie stosowanych technik, co zapewne wynika ze specyfiku rynku, dostępnych danych i ich typów.

Najczęściej stosowaną techniką uczenia maszynowego w tego typu badaniach jest uczenie nadzorowane. Jak wynika z pracy Ja’afar i in. (2021) w wielu badaniach, czy artykułach naukowych wykorzystywane są powtarzające się algorytmy tj. regresja liniowa, Gradient Boosting, sieci neuronowe, k-NN, SVM, drzewa decyzyjne czy lasy losowe. Ten ostatni model najczęściej wykazuje najlepsze predykcje cen nieruchomości. Popularność tych modeli wynika przede wszystkim z tego, że mają zdolność do kompleksowego odnalezienia zależności pomiędzy zmiennymi objaśniającymi a zmienną objaśnianą, w tym przypadku ceną (Dhar, Manikandan, 2023).

Regresja liniowa jest jednym z najprostszych modeli, który pokazuje relacje pomiędzy wartościami jednej zmiennej, a wielkością drugiej (Bruce i in., 2021). Proces ten jest właściwie jej zdaniem, które dokładnie polega na opisaniu, wyznaczeniu związku pomiędzy cechą objaśnianą a objaśniającą. Równanie modelu liniowego regresji jednowymiarowej można zapisać w następującej postaci:

We wzorze tym waga oznacza punkt przecięcia z osią *y*. Druga waga natomiast jest współczynnikiem wagowym zmiennej objaśniającej. Zadaniem regresji liniowej jako metody uczenia maszynowego jest uczenie wag modelu liniowego w sposób, który umożliwi opis związku pomiędzy zmiennymi, a ostatecznie na predykcję odpowiedzi nowych zmiennych objaśniających, które nie są częścią zestawu danych uczących. Istotą uczenia jest znalezienie najlepiej dopasowanej prostej przebiegającej pomiędzy punktami. Właśnie ta prosta jest nazywana linią regresji. Odcinki oznaczające odległość próbek od wspomnianej prostej wyznaczają błędy predykcji i są nazywane przesunięciami lub resztami (Raschka, Mirjalili, 2017).

Analiza regresyjna może być rozszerzona do wielu zmiennych objaśniających.   
W takim przypadku zastosowanie znajduje regresja wieloraka lub inaczej wielowymiarowa regresja liniowa. Model rozszerzany jest wtedy o wagi poszczególnych zmiennych, a równanie przyjmuje postać (Raschka, Mirjalili, 2017):

W momencie gdy w modelu jest wiele zmiennych niezależnych stosowana jest technika regresji LASSO, która opracowana została by rozwiązać problemy z nadmiernym dopasowaniem. Istotną własnością regresji LASSO jest dążenie do całkowitej eliminacji wag cech, które są najmniej istotne. Zadaniem algorytmu jest doprowadzenie do tzw. modelu rzadkiego. W zależności od siły regularyzacji wagi modelu mogą uzyskać wartość zerową. Sprawia to, że technika ta jest przydatna w nadzorowanym wyborze cech ((Raschka, Mirjalili, 2017). Algorytm LASSO jest określany inaczej jako kara stosowana do funkcji straty. Jak wspomniano sprawia, że model staje się prostszy, a tym samym zwiększa swoją zdolność do uogólniania. W przypadku metod regularyzacji najważniejszym jest parametr alfa, który jest siłą kary stosowanej za nadmierną złożoność modelu (https://machine-learning-and-data-science-with-python.readthedocs.io). Algorytm LASSO przyjmuje następującą postać (Szeliga, 2017):

W przypadku kiedy naruszone jest założenie liniowości pomiędzy zmiennymi objaśniającymi, a zmienną objaśniającą rozwiązaniem jest wykorzystanie regresji   
z wprowadzonym stopniem wielomianu. Taka technika nazywana jest regresją wielomianową, a równanie modelu przyjmuje postać (Raschka, Mirjalili, 2017):

Kolejną popularną metodą uczenia maszynowego wykorzystywaną w badaniach dotyczących predykcji cen nieruchomości są drzewa decyzyjne. Metoda ta inaczej nazywana drzewami klasyfikacji i regresji, drzewami losowymi lub też drzewami stanowi efektywną   
i powszechną metodę klasyfikacji oraz regresji. Stworzona została przez Breimana wraz   
z zespołem w 1984 roku. Z drzewami decyzyjnymi związane są jej metody pochodne tj. lasy losowe oraz drzewa wzmacniane. Wszystkie te algorytmy stanowią obecnie skuteczne narzędzia dla regresji i klasyfikacji. W odróżnieniu do regresji liniowej i logistycznej drzewa decyzyjne pokazują ukryte wzorce zachodzące w interakcjach pomiędzy danymi. W przypadku działania tego algorytmu można wyróżnić kilka kluczowych pojęć tj. podział rekursywny, kryterium podziału, węzeł, liść, strata, zanieczyszczenie, przycinanie. Pierwsze pojęcie oznacza wielokrotnie powtarzalny podział zbioru danych w celu utworzenia maksymalnie jednorodnych grup (Bruce i in., 2021). Powstawanie drzew decyzyjnych jest właśnie oparte przez rekurencyjne dzielnie danych treningowych na podzbiory (Szeliga, 2017). Dzielenie to odbywa się według kryterium podziału, czyli wartości predyktora dzielącego rekordy na mniejsze   
i większe od przyjętego kryterium. W drzewie decyzyjnym widoczne są węzły, które są graficzną lub regułową reprezentacją kryterium podziału. Liście natomiast to końce zbioru reguł. Pod pojęciem strata rozumie się liczbę błędnie zaklasyfikowanych rekordów na etapie podziału. Większa strata powoduje większe zanieczyszczenie, czyli rozszerzenie w pewnej części danych, w którym występuje mieszanina klas. Ostatnie pojęcie – przycinanie oznacza proces, podczas którego wykształcone już drzewo ma wstecznie przycinane gałęzie. Zadanie to ma na celu zredukowanie nadmiernego dopasowania (Bruce i in., 2021).

Jedną z odmian drzew decyzyjnych są lasy losowe, które można interpretować jako zbiór wielu drzew decyzyjnych z losowym próbkowaniem. Zadaniem tego algorytmu jest tak naprawdę redukcja wad drzew decyzyjnych. Jest to związane z tym, że lasy losowe ograniczają niestabilności drzew, a także tendencje do nadmiernego dopasowania do danych uczących. Odbywa się to dzięki uśrednianiu prognoz, które otrzymywane są z wielu drzew decyzyjnych (Cutler i in., 2012). Budowanie lasu losowego opiera się na kilku krokach. Pierwszym z nich jest ustalenie liczby drzew. Następnie algorytm wykorzystuje losowanie ze zwracaniem   
N obserwacji z N-elementowego zbioru. Proces ten odbywa się z założeniem, że prawdopodobieństwo wylosowania dla każdej próbki jest identyczne, czyli 1/N. Otrzymane obiekty są przenoszone do zbioru treningowego. Te elementy, które do niego nie trafiły trafiają natomiast do zbioru testowego. Trzecim krokiem jest losowanie bez zwracania ustalonej liczby zmiennych niezależnych. Następnie budowane jest drzewo, gdzie korzeniem jest zbiór wylosowany w etapie drugim i poszukiwane są najlepsze podziały dla zmiennych z kroku trzeciego. Następnie liczony jest błąd klasyfikacji i sprawdzenie czy zadana liczba drzew została osiągnięta. W przypadku gdy tak się nie stało model powraca do kroku drugiego, czyli rozpoczyna budowę nowego drzewa. Jeżeli liczba drzew jest satysfakcjonująca błąd klasyfikacji lasu losowego jest liczony jako średnia arytmetyczna błędów klasyfikacji wszystkich L drzew (Breiman, 2001).

Algorytm k-NN znajduje zastosowanie zarówno w przypadku problemów regresyjnych jak i klasyfikacyjnych. Uważany jest także za jeden z najprostszych i intuicyjnych nieparametrycznych klasyfikatorów. Ideą tego algorytmu jest znalezienie sposobu na opisanie podobieństw dwóch różnych próbek, a następnie predykcja dla nowej próbki. Przewidywanie to opiera się na przypisaniu jej wartości atrybutu decyzyjnego tej próbki, która jest do niej najbardziej podobna w aspekcie wartości atrybutów warunkowych. W przypadku problemów regresyjnych działanie algorytmu k-NN można opisać w następujących krokach (Król-Nowak, Kotarba, 2022):

1. Wybór sposobu mierzenia sąsiedztwa
2. Dobór hiperparametru k-liczby sąsiadów
3. Odnalezienie *k* obiektów, które są najbardziej podobne (leżące najbliżej) do nowego przykładu, dla którego należy określić wartość atrybutu decyzyjnego *y*
4. Obliczenie wartości średniej, medialny albo innej odpowiedniej miary statystycznej dla najbliższych sąsiadów
5. Wynik z poprzedniego punktu uznawany jest za wartość atrybutu decyzyjnego, który należy przypisać nowemu przykładowi

W zagadnieniach klasyfikacji powszechnie wykorzystywaną techniką jest regresja logistyczna. Służy ona do szacowania prawdopodobieństwa przynależności do konkretnej klasy. Wartością progową jest w tym przypadku zazwyczaj 50,00%. Wtedy model szacuje, że próbka należy do danej kategorii. W przeciwnym razie próbka nie jest zaliczana do określonej klasy. Model regresji logistycznej jest zatem klasyfikatorem binarnym (Geron, 2023). Koncepcja regresji logistycznej opiera się na ilorazie szans, czyli szansie wystąpienia danego zdarzenia (pozytywne zdarzenie). Termin ten mówi o zdarzeniu, którego wystąpienie chce się przewidzieć. Z ilorazami szans powiązana jest funkcja logitowa przyjmująca wartości od 0 do 1. Wykorzystywana jest do pokazania liniowego związku między wartościami cech,   
a zlogarytmowanymi szansami. Można to zilustrować za pomocą następującej funkcji (Raschka, Mirjalili 2017):

*gdzie p(y=1|x) – prawdopodobieństwo warunkowe, z którym dana próbka trafia do klasy 1 przy znanych cechach x*

Głównym zagadnieniem w przypadku regresji logistycznej jest prognozowanie prawdopodobieństwa przynależności próbki do określonej klasy, czyli odwrotność funkcji logitowej. Jest to więc sigmoidalna funkcja logistyczna przyjmująca postać (Raschka, Mirjalili, 2017):

W procesie budowy wszystkich opisanych powyżej algorytmów dane podzielono na zbiory treningowe oraz testowe w proporcji 80:20.

## 3.3. Miary oceny modeli

Modele oparte na zmiennych numerycznych zostały poddane ocenie jakości. W tym celu wykorzystano cztery powszechnie stosowane wskaźniki jakości tj. MAE, MSE, RMSE oraz R2.

Pierwsza miara, czyli średni błąd bezwzględny określa dokładność modeli regresyjnych. Im niższa wartość tego wskaźnika tym model jest trafniejszy. Równanie MAE można przedstawić za pomocą formuły (Szeliga, 2017):

Drugą miarą, którą wykorzystano w pracy był natomiast średni błąd kwadratowy (MSE). Jednak z uwagi na trudność w jego interpretacji za każdym razem obliczana była ocena pochodna, czyli pierwiastek błędu średniokwadratowego (RMSE). Zaletą tej miary jest możliwość porównywania wyników pomiędzy poszczególnymi modelami i łatwiejsza interpretacja. Podobnie jak poprzednio im mniejsza wartość RMSE, tym lepsze oszacowanie modelu (Szeliga, 2017).

*gdzie n – liczba obserwacji, i – kolejna obserwacja, a – rzeczywista wartość zmiennej wyjściowej i, p – oszacowana wartość zmiennej wyjściowej i*

Do oceny modeli posłużył jeszcze klasyczny współczynnik determinacji R2. Wskaźnik ten informuje jaka część zmienności zmiennej objaśnianej została wyjaśniona za pomocą zmiennych objaśniających. Innymi słowy jest miarą zdolności predykcyjnej modelu. Współczynnik determinacji przedstawia wariancję zmiennej wyjściowej w zależności od zmiennych wejściowych. W przypadku gdy model jest idealny suma kwadratów SSE jest równa 0, a wartość współczynnika osiąga 1. Z drugiej strony jeżeli model nie wykazuje siły predykcyjnej suma kwadratów błędów SSE jest równa sumie kwadratów różnic między wartościami rzeczywistymi a średnią SST. Wtedy miara ta osiąga wartość 0 (Szeliga, 2017).

*gdzie*

Ocena jakości modeli klasyfikujących opiera się na założeniu, że dla każdego przypadku znana jest wartość zmiennej wyjściowej. Należy więc poddać te rekordy klasyfikacji, a następnie porównać ze zwróconymi przez model. Odbywa się to przez zliczenie rekordów właściwie i niewłaściwe skategoryzowanych przez model. W ten sposób konstruowana jest macierz pomyłek składająca się z dwóch kolumn i dwóch wierszy oznaczonych symbolami *TP, T, FP, FN*. Pierwszy symbol oznacza przypadki *True Positives,* czyli liczbę pozytywnych przypadków poprawnie zaklasyfikowanych do klasy pozytywnej. Druga miara (*False Negatives)* to z kolei liczba przypadków pozytywnych, ale niepoprawnie zaklasyfikowanych jako należące do klasy negatywnej. Błąd ten nazywany jest błędem drugiego rodzaju. Liczba negatywnych przypadków, które zostały niepoprawnie skategoryzowane jako należące do klasy pozytywnej stanowią z kolei błąd pierwszego rodzaju i określane są jako FP (*False Positives).* Ostatni element macierzy pomyłek to przypadki *True Negatives*, czyli negatywne, które zostały dobrze sklasyfikowanej jako należące do klasy negatywnej (Szeliga, 2017).

Macierz pomyłek stanowi podstawę do obliczenia ogólnych miar jakości modeli klasyfikujących. Powszechnie wykorzystuje się następujące miary:

1. Dokładność (*accuracy*) – mierzy proporcję prawidłowy klasyfikacji do wszystkich klasyfikacji – *(TP + TN) / (TP + FN + FP +TN)*
2. Precyzja (*precision*) – w tym przypadku można wyróżnić dwa pomiary. Wynikiem pierwszej z nich jest proporcja prawidłowych pozytywnych klasyfikacji, czyli ile pozytywnie zaklasyfikowanych przypadków zostało poprawnie skategoryzowanych *(TP / TP + FP).* Analogicznie można wyznaczyć ufność dla negatywnych przewidywań, czyli to w jakim procencie predykcje negatywne rzeczywiście są negatywne *(TN / TN + FN)*
3. Czułość (*recall*) – dokonuje pomiaru proporcji poprawnych klasyfikacji względem wszystkich pozytywnych przypadków. Odpowiada więc na pytanie, ile z pozytywnych przypadków jest dobrze zaklasyfikowanych *(TP / TP + FN)*
4. Specyficzność (*specifity*) – nazywana jest miarą *True Negatives* lub znamiennością. Wyraża ona proporcję prawidłowych negatywnych klasyfikacji względem wszystkich negatywnych przypadków – ile z negatywnych przypadków zostało dobrze zaklasyfikowanych *(TN / TN + FP)*

Tabela 2. Macierz pomyłek

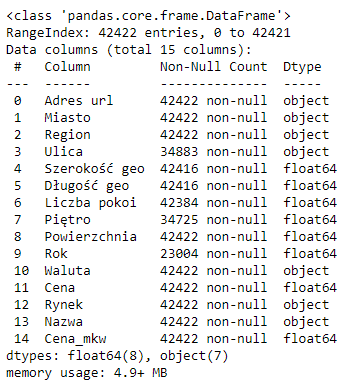
|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Wartości przewidywane** | |
| **Negatywne** | **Pozytywne** |
| **Wartości rzeczywiste** | **Negatywne** | True Negative (TN) | False Positive (FP) |
| **Pozytywne** | False Negative (FN) | True Positive (TP) |

Źródło: Opracowanie własne na podstawie www.medium.com

Jakość modeli klasyfikujących można również przedstawić w postaci graficznej.   
W tym przypadku najbardziej znanym narzędziem jest krzywa ROC. Jest to wykres, gdzie na osi odciętych znajduje się 1 – specyficzność, a na osi rzędnych czułość. Punkt w tej przestrzeni reprezentujący klasyfikator przedstawia kompromis pomiędzy współczynnikiem trafień (zysk), a współczynnikiem fałszywych alarmów (koszt). Pozwala to znaleźć odpowiedź na pytanie jak często model klasyfikował prawidłowo pozytywne przypadki. Przy ocenie modelu powszechnie wykorzystuje się pole powierzchni pod krzywą ROC, czyli obszar AUC.   
W przypadku modelu idealnego wartość ta osiąga 1,00 i jest to model doskonały. Pole AUC dla klasyfikatora losowego wynosi natomiast 0,50 i jest to wartość poniżej, której model jest nieakceptowalny.

# 4. Eksploracyjna analiza danych

Pobrane z serwisu ogłoszeniowego dane zapisane zostały w postaci *data frame*   
z wykorzystaniem biblioteki Pandas. Tak wczytany zbiór danych obejmował 42 422 rekordów i 14 kolumn, czyli łącznie 593 908 danych. Kolumny tabeli składały się z następujących zmiennych: *adres url, miasto, region, ulica, szerokość i długość geograficzna, liczba pokoi, piętro, powierzchnia, rok, waluta ceny, cena, rynek, tytuł ogłoszenia*. Na tym etapie dodano jeszcze jedną zmienną, która reprezentowała cenę za metr kwadratowy. Wczytane cechy były reprezentowane przez określone typy, na temat których informacje uzyskano za pomocą metody .*info()*. Na początku pięć zmiennych reprezentowało dane numeryczne typu liczb zmiennoprzecinkowych *float64* i były to zmienne typowo ilościowe tj. powierzchnia, liczba pokoi, długość i szerokość geograficzna oraz nowo utworzona cena za metr kwadratowy. Największa liczba kolumn zawierała zmienne typu *object* (9) i reprezentowały cechy jakościowe – adres url, miasto, region, ulica, piętro, rok budowy budynku, waluta ceny, rynek oraz tytuł ogłoszenia. Zmienna dotycząca cen mieszkań reprezentowana była przez format *int64*, jednak w celu dalszej analizy przekształcona została na liczby zmiennoprzecinkowe. Podobnie postąpiono w przypadku piętra mieszkania oraz roku budynku. Ostatecznie więc zbiór danych liczył osiem zmiennych liczbowych (*float64*) oraz siedem zmiennych typu *object*.



Rysunek 2. Zmienne oraz ich typy pobrane w wyniku web scraping

Źródło: Opracowanie własne

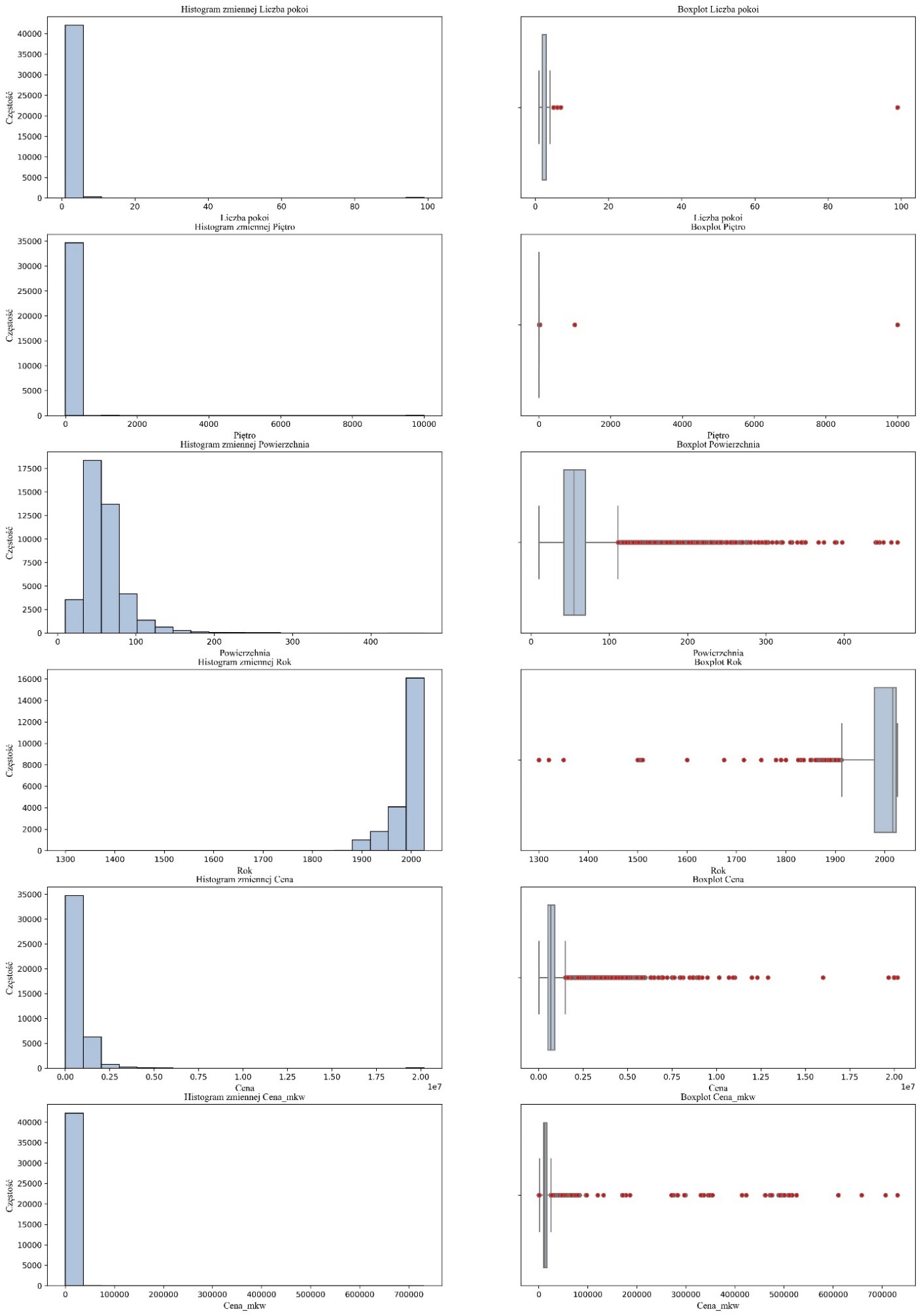
Analiza braków danych wykazała, że najwięcej brakujących informacji dotyczyło roku budowy budynku. W tym przypadku informacja nie została pobrana dla 19 418 ofert, co stanowiło prawie 46,00% całkowitej liczby ofert. Braki danych dotyczyły również zmiennych piętro (18,14%) oraz adresu piętro (17,77%). W przypadku pozostałych zmiennych pozyskano wszystkie niezbędne informacje. Niektóre spośród zmiennych nie niosły cennych informacji   
z perspektywy dalszej analizy dlatego też w kolejnym kroku cechy tj. adres url, region, waluta ceny zostały usunięte.

Kolejnym krokiem eksploracyjnej analizy danych było podsumowanie w postaci statystyk opisowych. Etap ten pozwolił na stwierdzenie, że w zebranych danych występują wartości odstające, a rozkłady tych zmiennych charakteryzują się dużą skośnością (rys. 3). Wynikało to przeważnie z błędnie wprowadzonych informacji do ogłoszeń. W przypadku liczby pokoi maksymalna wartość wynosiła 99, co jednoznacznie jest wartością błędną. Podobna sytuacja miała miejsce dla zmiennej oznaczającej piętro mieszkania, gdzie ogłoszenie zawierało wartość 9999. Innym przykładem może być rok, dla którego wartość minimalna wynosiła 1300. W związku z tym przystąpiono do identyfikacji wartości odstających oraz ich usunięcia, co ostatecznie spowodowało, że liczba rekordów przyjęta do dalszej analizy zmniejszyła się do 35 346. Identyfikację przypadków odstających przeprowadzono za pomocą formuły:

Tabela 3. Statystyki opisowe poszczególnych zmiennych przed usunięciem outlierów

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Statystyki** | **Liczba**  **pokoi** | **Piętro** | **Powierzchnia** | **Rok** | **Cena** | **Cena\_mkw** |
| **Liczba** | 42 384 | 34 725 | 42 422 | 23 004 | 42 422 | 42 422 |
| **Średnia** | 3 | 19 | 60 | 1997 | 855 486 | 14 494 |
| **Odchylenie std** | 4 | 398 | 29 | 37 | 1 095 127 | 20 467 |
| **Minimum** | 1 | -2 | 10 | 1300 | 10 000 | 182 |
| **Q1** | 2 | 1 | 42 | 1979 | 515 755 | 10 152 |
| **Mediana** | 3 | 3 | 55 | 2016 | 684 000 | 12 744 |
| **Q3** | 3 | 4 | 70 | 2023 | 910 000 | 16 088 |
| **Max** | 99 | 9 999 | 468 | 2026 | 20 190 000 | 731 261 |

*Źródło:* Opracowanie własne



Rysunek 3. Histogramy oraz wykresy pudełkowe przed usunięciem outlierów

*Źródło:* Opracowanie własne

Opisane powyżej usunięcie wartości odstających spowodowało zmianę rozkładów poszczególnych zmiennych numerycznych, a tym samym ich statystyk opisowych. Wyniki przedstawione zostały w tabeli 3.

Tabela 4. Statystyki opisowe zmiennych po usunięciu obserwacji odstających

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Statystyki** | **Liczba pokoi** | **Piętro** | **Powierzchnia** | **Rok** | **Cena** | **Cena\_mkw** |
| **Liczba** | 35 319 | 28 797 | 35 346 | 18 002 | 35 346 | 35 346 |
| **Średnia** | 2 | 3 | 53 | 2003 | 685 004 | 13 234 |
| **Odchylenie std** | 1 | 2 | 17 | 27 | 245 215 | 4 168 |
| **Minimum** | 1 | 1 | 10 | 1917 | 113 000 | 6 002 |
| **Q1** | 2 | 1 | 41 | 1985 | 501 665 | 10 284 |
| **Mediana** | 2 | 3 | 52 | 2018 | 650 000 | 12 581 |
| **Q3** | 3 | 4 | 64 | 2023 | 834 000 | 15 452 |
| **Max** | 4 | 8 | 106 | 2026 | 1 403 360 | 40 000 |

*Źródło:* Opracowanie własne

Średnio na każde mieszkanie w analizowanym zbiorze przypadały dwa pokoje. Minimalnie liczba pokoi wynosiła 1. Kwartyl pierwszy oraz mediana w przypadku tej zmiennej wyniosły 2, co oznacza, że połowa mieszkań posiadała mniej lub więcej niż 2 pokoje. Kwartyl trzeci natomiast wynosił 3, czyli 75,00% nieruchomości miało mniej niż 3 pokoje, a 25,00% więcej pomieszczeń. Mieszkania pochodzące z przyjętych do analizy ogłoszeń posiadały maksymalnie cztery pomieszczenia.

Lokale mieszkalne z analizowanego zbioru średnio położone były na trzecim piętrze. Większość mieszkań (75,00%) zlokalizowanych było na niższych niż czwarta kondygnacja. Reszta z nich to mieszkania położone na wyższych piętrach. Lokale położone najwyżej mieściły się na ósmym piętrze.

Średnia wielkość mieszkań z pobranych ogłoszeń wynosiła 53,38 m2. Odchylenie standardowe dla tej cechy wynosiło 16,61 m2, co oznacza, że średnio o tyle wartości różniły się od wartości przeciętnej i wskazuje na umiarkowane rozproszenie danych – współczynnik zmienności na poziomie 32,00%. Najmniejsze mieszkanie ze zbioru miało jedynie 10,00 m2, a największe natomiast 106,17 m2. Wartość mediany była bliska wartości średniej i sugeruje, że połowa mieszkań była mniejsza niż 52 m2. Kwartyl trzeci w tym przypadku wynosił 64,24 m2, co oznacza, że jedna czwarta ogłoszeń dotyczyła mieszkań większych niż wspomniana powierzchnia.

Pobrane oferty dotyczyły przeważnie mieszkań mieszczących się w budynkach z 2003 roku. Większość ofert była związana z lokalami w budynkach młodszych niż 2023 rok, a jedna czwarta w obiektach, których rok budowy miał miejsce później niż w 2023 roku. Wartość maksymalna wynosząca 2026 sugeruje, że w zbiorze znalazły się ogłoszenia z rynku pierwotnego z planowanymi datami do użytkowania.

Wartość przeciętna w przypadku najważniejszej cechy, czyli cen mieszkań wynosiła ponad 685 003,65 złotych. Zmienna ta charakteryzowała się również największym rozproszeniem wartości. Odchylenie standardowe wynosiło bowiem ponad 245 214,80 złotych. Współczynnik zmienności dla cen mieszkań wynosił zatem ponad 35,00%. Najniższa cena ofertowa w analizowanym zbiorze kształtowała się na poziomie 113 000 złotych. Najdroższe mieszkanie kosztowało natomiast ponad 1 403 360 złotych. Jak wynika z interpretacji kwartyla pierwszego większość mieszkań (75,00%) stanowiły nieruchomości o cenie ponad 501 665,25 złotych. Najdroższe mieszkania stanowiące 25,00% całego zbioru kosztowały ponad 834 000 złotych.

Najniższa cena w przeliczeniu na metr kwadratowy wynosiła natomiast 6 002,09 złotych, najwyższa natomiast ponad 40 000 złotych. Średnio mieszkania z analizowanego zbioru kosztowały 13 234 tys. złotych za metr kwadratowy, a większość mieszkań (75,00%) stanowiły mieszkania, których cena kształtowała się na poziomie poniżej 15 452,38 złotych za metr kwadratowy. Przeciętnie wartości w przypadku tej zmiennej odchylały się od średniej   
o 4 167,51 tys. złotych, co wskazuje na umiarkowane rozproszenie tej cechy (31,00%).

Obraz zawierający tekst, diagram, zrzut ekranu, Równolegle

Opis wygenerowany automatycznie

Rysunek 4. Histogramy oraz wykresy pudełkowe po usunięciu outlierów

*Źródło:* Opracowanie własne

Usunięcie wartości odstających pozwoliło na bardziej prawidłową analizę korelacji pomiędzy zmiennymi ilościowymi (rys. 5). Spośród wszystkich zmiennych największą współzależnością charakteryzowały się liczba pokoi oraz powierzchnia (0,82), co oznacza dość silną korelację. Wartość ta jest dodatnia co oznacza, że wraz ze wzrostem jednej zmiennej rosną również wartości drugiej. Cena mieszkania najsilniej korelowała z powierzchnią lokali. Wartość współczynnika korelacji liniowej w tym przypadku wynosiła 0,6. Podobnie jak poprzednio związek ten był dodatni. W nieco mniejszym stopniu (0,48) zmienność cen mieszkań była wyjaśniania przez liczbę pokoi oraz cenę za metr kwadratowy (0,53). Pozostałe współczynniki dla par zmiennych osiągnęły niskie wartości oznaczające brak korelacji.

Obraz zawierający tekst, zrzut ekranu, kwadrat, Prostokąt

Opis wygenerowany automatycznie

Rysunek 5. Macierz korelacji dla zmiennych numerycznych

*Źródło:* Opracowanie własne

Rozpatrując powiązanie cen mieszkań ze zmienną jakościową jaką jest miasto można zauważyć, że najdroższe mieszkania występowały w Warszawie, gdzie średnia cena ofertowa wynosiła ponad 833 908 złotych. Na drugim miejscu znalazł się Kraków, w którym przeciętna stawka ofertowa wynosiła prawie 800 000 złotych. Średnia cena mieszkania była najniższa   
w takich miastach wojewódzkich jak Bydgoszcz, Zielona Góra, Łódź i Gorzów Wielkopolski. Dla tego ostatniego miasta przeciętna cena ofertowa wynosiła 428 136 złotych.

Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie

Rysunek 6. Średnie ceny ofertowe mieszkań w poszczególnych miastach

*Źródło:* Opracowanie własne

Wśród zmiennych znajdowała się również druga cecha nienumeryczna oznaczająca rynek wtórny lub pierwotny. Wśród wszystkich ogłoszeń większość stanowiły mieszkania pochodzące z rynku pierwotnego, których było 21 133. Stanowiło to 60,00% całego zbioru danych. Pozostałe oferty (14 213) pochodziły z rynku wtórnego. Średnie ceny w podziale na rodzaj rynku są do siebie bardzo zbliżone (rys. 7). Mieszkania pochodzące z rynku wtórnego są jednak przeciętnie droższe (693 992 złotych) niż z pierwotnego (678 959 złotych).

Obraz zawierający tekst, zrzut ekranu, Prostokąt, diagram

Opis wygenerowany automatycznie

Rysunek 7. Średnie ceny ofertowe w podziale na rodzaj rynku

*Źródło:* Opracowanie własne

# 5. Wyniki modelowania

## 5.1. Regresja liniowa

Pierwszym modelem uczenia maszynowego wykorzystanym w pracy była regresja liniowa dla pary zmiennych cena mieszkania oraz powierzchnia, które wykazały najsilniejszą współzależność. Graficzna interpretacja tego związku (rys. 8) pozwala stwierdzić, że linia regresji odpowiada trendowi danych. Współczynnik nachylenia dla tego modelu wyniósł 8 857,90. Oznacza to, że wraz z jednostkowym wzrostem powierzchni o tyle wzrośnie cena mieszkania. Wyraz wolny z kolei osiągnął wartość 211 881,12.

Obraz zawierający zrzut ekranu, linia, stok, tekst

Opis wygenerowany automatycznie

Rysunek 8. Model regresji liniowej

*Źródło:* Opracowanie własne

Średni błąd bezwzględny (MAE) modelu regresji liniowej dla pary zmiennych cena   
i powierzchnia wynosił 155 024,01, co oznacza, że średnia wartość bezwzględna różnic między prognozowanymi, a rzeczywistymi wartościami była duża. Potwierdzeniem słabej jakości modelu mogą być również wartości miar MSE i pochodnego RMS (194 858,20). Interpretując współczynnik determinacji R2 można stwierdzić, że zmienność cen mieszkań została wyjaśniona zmiennością powierzchni lokali mieszkalnych w zaledwie 36,00%. Potwierdzeniem słabej jakości modelu jest również wykres reszt regresji (rys. 9

), na którym można zauważyć, że nie układają się one w sposób losowy.

Obraz zawierający tekst, diagram, zrzut ekranu, Wykres

Opis wygenerowany automatycznie

Rysunek 9. Reszty regresji liniowej

*Źródło:* Opracowanie własne

## 5.2. Regresja wielomianowa

W dalszej kolejności przeprowadzono modelowanie z wykorzystaniem regresji wielomianowej, która wykorzystywana jest głównie przy naruszeniu złożenia liniowości. Jednym z ważniejszych problemów w tym modelu może być wybór stopnia wielomianu wykorzystanego do jego budowy. Wyniki uwzględniające wielkość błędu MSE w zależności od wielkości wielomianu (rys. 10) pokazują, że w przypadku zmiennych cena mieszkania   
i powierzchnia optymalne wyniki plasują się pomiędzy 4 a 10 wielomianem. Najniższą wartość uzyskano jednak przy wielomianie równym 8.

Tak skonstruowany model (rys. 11). charakteryzował się bardzo zbliżonymi wartościami błędów co regresja liniowa, jednak wszystkie były niższe. Błąd MAE w tym przypadku wyniósł 154 535,92, a RMSE 194 210,12. Dla regresji wielomianowej uzyskano także nieznacznie wyższy współczynnik determinacji (37,13%).

Obraz zawierający tekst, linia, Wykres, diagram

Opis wygenerowany automatycznie

Rysunek 10. Stopień wielomianu dla regresji wielomianowej w porównaniu do błędu MSE

*Źródło:* Opracowanie własne

Obraz zawierający zrzut ekranu, tekst, Wykres, stok

Opis wygenerowany automatycznie

Rysunek 11. Model regresji wielomianowej ze stopniem wielomianu równym 6

*Źródło:* Opracowanie własne

## 5.3. Regresja wieloraka i LASSO

Regresja liniowa pomiędzy wspomnianą parą najsilniej skorelowanych zmiennych nie przyniosła modelu dobrej jakości. W związku z tym kolejną techniką uczenia maszynowego wykorzystaną w pracy była regresja wieloraka. W tym celu jako zmienne objaśniające wykorzystano następujące zmienne numeryczne: liczba pokoi, piętro, powierzchnia, cena za metr kwadratowy, odległość od centrum miasta. Współczynniki poszczególnych zmiennych przedstawiono w tabeli 4. Otrzymany model posiadał znacznie niższe wartości błędów niż poprzednie (MAE 44 147,67 a RMSE 64 793,09). Zmienność zmiennych objaśniających   
w bardzo wysokim stopniu (93,00%) wyjaśniała zmienność cen mieszkań.

Tabela 5. Współczynniki regresji wielorakiej dla poszczególnych zmiennych

|  |  |
| --- | --- |
| **Zmienna** | **Współczynnik regresji** |
| Liczba pokoi | 13 586,61 |
| Piętro | 1 779,07 |
| Powierzchnia | 11 932,89 |
| Cena za metr | 46,73 |
| Odległość od centrum | -12,10 |

*Źródło:* Opracowanie własne

Interpretując otrzymane wyniki można stwierdzić, że każdy dodatkowe pomieszczenie   
w mieszkaniu podniesie wartość mieszkania o 13 586,61 złotych. W przypadku piętra, na której mieści się lokal mieszkalny gdy numer kondygnacji wzrasta o 1 cena mieszkania zmienia się   
o 1 779,07 złotych. Dodatkowa jednostka powierzchni nieruchomości sprawia, że może być ono droższe o 11 932,89 złotych. Rozpatrując mieszkanie pod kątem lokalizacji okazuje się, że dodatkowy kilometr odległości od centrum miasta sprawia, że mieszkanie tanieje o 12,10 złotych.

Wykres reszt dla modelu regresji wielorakiej (rys. 12) pokazuje, że są one rozłożone mniej więcej równomiernie na całej długości. Może to sugerować, że model nie wystąpiły problemy z modelem i danymi.

Obraz zawierający tekst, zrzut ekranu, linia

Opis wygenerowany automatycznie

Rysunek 12. Reszty regresji wielorakiej

*Źródło:* Opracowanie własne

Z uwagi na możliwość zbyt dobrego dopasowania danych do modelu i zbyt dużą liczbę zmiennych na tym etapie przeprowadzona została dodatkowo regresję metodą LASSO. Uwzględnia ona kary za dużą liczbę cech objaśniających. Najlepszym parametrem dla tej metody okazała się wartość 1, przy której RMSE dla najkorzystniejszego modelu wyniosło 64 973, 12, natomiast R2 prawie 93%. Sugeruje to, że model regresji wielorakiej był dobrze dopasowany do danych, a oba modele dobrze przewidywały zmienność w danych treningowych.

## 5.4. Regresja ze zmiennymi kategorialnymi

Uzyskane wyniki skłoniły do włączenia do analizy zmiennych jakościowych   
i zbudowania modelu regresji ze zmiennymi kategorycznymi. Model ten zawierał łącznie   
7 zmiennych objaśniających. Zmienne jakościowe były reprezentowane przez miasto oraz rynek. Błędy MAE, MSE oraz RMSE uległy obniżeniu i wynosiły odpowiednio 43 230,58, 4 054 837 245,67 i 63 677,60. W przypadku modelu regresji ze zmiennymi jakościowymi współczynnik R2 osiągnął najwyższą wartość równą 93,24%. Ocena reszt modelu (rys. 13) pokazuje, że przewidywane wartości są skoncentrowane wokół linii przekątnej, co sugeruje bardzo dobre dopasowanie modelu do danych. Potwierdzeniem tego może być również histogram błędów predykcji (rys. 14) przypominający rozkład normalny. Błędy predykcji są rozmieszczone prawie symetrycznie względem zera. Oznacza to, że model jest dobrze przewiduje wartości powyżej oraz poniżej wartości rzeczywistych.

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Rysunek 13. Model regresji ze zmiennymi kategorialnymi

*Źródło:* Opracowanie własne

Obraz zawierający tekst, diagram, zrzut ekranu, Wykres

Opis wygenerowany automatycznie

Rysunek 14. Histogram reszt regresji ze zmiennymi kategorialnymi

*Źródło:* Opracowanie własne

Patrząc na reszty modelu pod kątem zmiennych jakościowych (rys. 15) można zauważyć, że model dobrze radzi sobie z przewidywaniem cen mieszkań w poszczególnych miastach – wykresy pudełkowe reszt są podobne. Zauważalną różnicą są obserwacje odstające w przypadku Warszawy, co może sugerować większe zróżnicowanie cen mieszkań w stolicy,   
a tym samym większe trudności w przewidywaniu cen przez model. Analogiczna sytuacja występuje w przypadku podziału danych na rynek, gdzie większe trudności może mieć model dla rynku wtórnego.

Obraz zawierający tekst, diagram, Plan, Wykres

Opis wygenerowany automatycznie

Rysunek 15. Reszty regresji ze zmiennymi kategorialnymi w podziale na miasta

*Źródło:* Opracowanie własne

Obraz zawierający diagram, zrzut ekranu, linia, Prostokąt

Opis wygenerowany automatycznie

Rysunek 16. Reszty regresji ze zmiennymi kategorialnymi w podziale na rodzaj rynku

*Źródło:* Opracowanie własne

## 5.5. Drzewa decyzyjne i lasy losowe - regresja

Kolejną techniką uczenia maszynowego wykorzystaną w pracy były drzewa decyzyjne. Przedmiotem modelowania były zmienne numeryczne wspomniane we wcześniejszych metodach, a zmienną objaśnianą ceny mieszkań. Parametry modelu drzewa decyzyjnego wynosiły: maksymalna głębokość 3, minimalna liczba próbek do utworzenia liścia 20 oraz minimalna liczba próbek wymagana do podziału węzła na poziomie 30.

Model ten charakteryzował się bardzo dobrymi wartościami błędów (tab. 5) oraz wysoką zdolnością wyjaśniania zmienności (92,00%). Wyniki te były jednak gorsze   
w porównaniu do regresji ze zmiennymi kategorialnymi. W związku z tym skorzystano   
z metody automatycznego doboru parametrów opartej na przeszukiwaniu siatki w połączeniu   
z walidacją krzyżową. Tak uzyskane parametry (max\_depth = 7, min\_samples\_lead = 1, min\_samples\_split = 2) pozwoliły na uzyskanie modelu znacznie lepszego niż poprzedni.   
W tym przypadku wszystkie błędy zostały obniżone, a współczynnik determinacji osiągnął wartość 97,77%.

Tabela 6. Parametry i oceny jakości dla modelu drzew decyzyjnych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parametry drzewa** | **MAE** | **MSE** | **RMSE** | **R^2** |
| max\_depth = 7, min\_samples\_lead = 1, min\_samples\_slit = 2 | 54724,59 | 4881100789,65 | 69864,88 | 91,86% |
| max\_depth = 3, min\_samples\_lead = 20, min\_samples\_slit = 30 | 28380,94 | 1340579831,11 | 36613,93 | 97,77% |

*Źródło:* Opracowanie własne

Z wizualizacji drzewa decyzyjnego (rys. 17) można wyczytać warunki decyzyjne, błędy i statystyki dotyczące liścia. Przykładowo w pierwszym kroku zbudowanego modelu zbiór danych jest dzielony pod kątem powierzchni (54,99 m2). Po takim podziale do liścia trafiło 2 876 próbek na podstawie warunku decyzyjnego. Prognozowana wartość dla mieszkań poniżej wspomnianego progu wyniosła 685 010 złotych. W drugim podziale zbiory są dzielone pod kątem ceny za metr kwadratowy.

Warto również dodać, że analiza atrybutu feature impotances wykazała istnienie hedynie dwóch ważnych zmiennych przy budowie drzewa decyzyjnego – powierzchnia (46,43%) oraz cena za metr kwadratowy (54,57%). Pozostałe zmienne nie zostały wykorzystane do budowy drzewa decyzyjnego.

Obraz zawierający tekst, linia, diagram

Opis wygenerowany automatycznie

Rysunek 17. Model drzew decyzyjnych

*Źródło:* Opracowanie własne

Pochodnym algorytmem w stosunku do drzew decyzyjnych są lasy losowe. Wyniki ważności cech do budowy modelu były analogiczne jak w poprzednim przypadku. Jednak   
w przypadku lasów losowych błędy oceny modelu były najniższe spośród wszystkich technik uczenia maszynowego. Średni błąd bezwzględny w tym przypadku wynosił 1 717,55, średni błąd kwadratowy (MSE) 15 372 193,01, a jego pierwiastek (3 920,74). Model charakteryzował się również najwyżej współczynnikiem determinacji (99,97%). Oznacza to, że zmienność cen mieszkań została wyjaśniona niemal całkowicie zmiennością cech objaśniających.

W przypadku lasów losowych podział drzewa początkowo wyglądał bardzo podobnie jak w przypadku drzew decyzyjnych. Jednak jak można zauważyć (rys. 18) w drugim kroku zbiór został podzielony pod kątem powierzchni mieszkań, a nie ceny za metr kwadratowy. Dla pierwszego liścia przyjęty został próg podziału na poziomie 40,95 m2, a dla drugiego 70,51 m2.

Obraz zawierający tekst, linia, diagram

Opis wygenerowany automatycznie

Rysunek 18. Model lasów losowych

*Źródło:* Opracowanie własne

## 5.6. Algorytm k-NN

Ostatnim algorytmem wykorzystanym w zagadnieniach regresyjnych był k-NN, którego najlepszy hipeparametr określono za pomocą testowania wartości liczby sąsiadów   
i jego wpływu na błąd RMSE. Jak można zauważyć (rys. 19) najmniejszy pierwiastek błędu średniokwadratowego ma miejsce dla liczby sąsiadów równych 14.

Model k-NN z takim hiperparametrem cechował się bardzo wysoką jakością. Współczynnik determinacji pokazuje, że poprzez zmienność lokalizacji można wyjaśnić 99,00% zmienności cen mieszkań. Pierwiastek błędu średniokwadratowego dla modelu wynosił 24 202, a błąd absolutny 12 078,03. Uzyskane prawidłowości można dostrzec na rysunku (rys. 20), na którym widoczne jest prawie idealne dopasowanie modelu do danych. Z uwagi na tak wysoki wynik istniała obawa, że algorytm może być przeuczony. W związku z tym przeprowadzona została walidacja krzyżowa. Średni wynik współczynnika determinacji był jednak bardzo zbliżony do wcześniej uzyskanej wartości (96,75%). Oznacza to, że model dobrze generalizuje dane oraz dobrze radzi sobie z danymi treningowymi jak i nowymi. Nie można w takim wypadku mówić o przeuczeniu modelu.

Obraz zawierający tekst, linia, Wykres, diagram

Opis wygenerowany automatycznie

Rysunek 19. Liczba sąsiadów algorytmu k-NN w porównaniu z błędem RMSE

*Źródło:* Opracowanie własne

Obraz zawierający zrzut ekranu, tekst, linia, Wykres

Opis wygenerowany automatycznie

Rysunek 20. Model k-NN

*Źródło:* Opracowanie własne

## 5.7. Regresja logistyczna

Pierwszym algorytmem klasyfikującym była regresja logistyczna. Zadaniem modelu była na podstawie zmiennych numerycznych (liczba pokoi, piętro, powierzchnia, cena za metr kwadratowy, odległość od centrum) predykcja czy mieszkanie jest zlokalizowane w Warszawie. Model ten charakteryzował się dokładnością (accuracy) na poziomie 78,49%, co oznacza, że taka część przypadków została zaklasyfikowana poprawnie. Model nie miał również problemu z określeniem, które mieszkanie nie znajduje się w stolicy (81,00%). Większą trudność można zauważyć w przypadku gdy miasto było Warszawą i model rzeczywiście zaklasyfikował mieszkanie do tej klasy. Precyzja w tej kategorii wynosiła bowiem 52%. Potwierdzają to również wyniki czułości (18,00%) oraz specyficzności (95,00%).

Tabela 7. Macierz pomyłek dla modelu regresji logistycznej

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategoria** | **Precision** | **Recall** | **Accuracy** |
| TN (nie Warszawa) | 0,81 | 0,95 | 0,78 |
| TP (Warszawa) | 0,52 | 0,18 |

*Źródło:* Opracowanie własne

Jak można się było spodziewać zmiennymi, które mogą polepszyć jakość modelu   
w tym przypadku były współrzędne geograficzne. Dokładność modelu (*accuracy*)   
z dodatkowymi zmiennymi wyniosła 92,96%. Zdecydowanej poprawie uległa miara precyzji przy poprawnym klasyfikowaniu Warszawy (82%). Model niemal dokładnie rozpoznawał także, że miasto nie jest tym miastem (96%).

Tabela 8. Macierz pomyłek dla modelu regresji logistycznej z dodatkowymi zmiennymi

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategoria** | **Precision** | **Recall** | **Accuracy** |
| TN (nie Warszawa) | 0,96 | 0,95 | 0,93 |
| TP (Warszawa) | 0,82 | 0,87 |

*Źródło:* Opracowanie własne

Zaobserwowane prawidłowości są również dobrze widoczne na krzywej ROC, na której zawarte zostały oba modele (rys. 21). Krzywa ta dla modelu ze współrzędnymi geograficznymi zbliża się do osi pionowej co oznacza wysoką czułość, przy niskim współczynniku fałszywie pozytywnych kategorii. Pole pod krzywą dla tego modelu wyniosło 0,96 i jest bliskie 1,00, czyli sytuacji idealnej separacji między klasami.

Obraz zawierający tekst, linia, Wykres, diagram

Opis wygenerowany automatycznie

Rysunek 21. Krzywa ROC dla dwóch modeli regresji logistycznej

*Źródło:* Opracowanie własne

## 5.8. Drzewa decyzyjne i lasy losowe - klasyfikacja

Interesującym zagadnieniem może być także klasyfikowanie mieszkań ze względu na rynek. Na podstawie zmiennych numerycznych określających liczbę pokoi, piętro, powierzchnię, cenę za metr kwadratowy, odległość od centrum oraz ceny starano się prognozować to czy mieszkanie pochodzi z tynku wtórnego czy pierwotnego.

Model regresji logistycznej dla przyjętych zmiennych (tab. 8) charakteryzował się dokładnością na poziomie 60,00%. Można jednak zauważyć, że model miał trudności   
w rozpoznawaniu rodzaju rynku. Ponad połowa (59,00%) przypadków zaklasyfikowanych jako rynek wtórny rzeczywiście było mieszkaniami używanymi. Jednak to stanowi jedynie 5,00% wszystkich dobrze zaklasyfikowanych przypadków (*recall*). Bardzo wysoka wrażliwość (98,00%) sugeruje, że model bardzo dobrze radzi sobie z identyfikacja mieszkań z rynku pierwotnego. Takie wyniki mogą być efektem tego, że rodzaj rynku to zmienna posiadająca tylko dwie kategorie z przewagą ofert z rynku pierwotnego.

Tabela 9. Macierz pomyłek dla modelu regresji logistycznej klasyfikujący rodzaj rynku

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategoria** | **Precision** | **Recall** | **Accuracy** |
| TN (nie pierwotny) | 0,59 | 0,05 | 0,60 |
| TP (pierwotny) | 0,6 | 0,98 |

*Źródło:* Opracowanie własne

Niewiele lepsze wyniki przy tym zagadnieniu uzyskał model drzew decyzyjnych   
z dokładność na poziomie 64,60% ('max\_depth': 3, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2). Zdecydowaną różnicą, którą można zauważyć był wzrost wrażliwości (*sensitivity*) do wartości 24,00%. Oznacza to, że tym razem model tyle poprawnie zaklasyfikował mieszkań   
z rynku wtórnego spośród wszystkich obiektów należących do tej klasy. W tym przypadku mamy również do czynienia ze spadkiem specyficzności (*specifity*). Wynik ten oznacza, że 92,00% wszystkich mieszkań z rynku pierwotnego rzeczywiście należących do tej klasy zostało poprawnie skalsyfikowanych.

Tabela 10. Macierz pomyłek dla drzewa decyzyjnego - predykcja rynku

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategoria** | **Precision** | **Recall** | **Accuracy** |
| TN (nie pierwotny) | 0,67 | 0,24 | 0,65 |
| TP (pierwotny) | 0,64 | 0,92 |

*Źródło:* Opracowanie własne

Przebieg podziału drzewa decyzyjnego (rys. 22) pokazuje, że pierwszy zbiór został podzielony pod kątem ceny za metr kwadratowy (7 889,34 zł). W drugim podziale na lewy i prawy korzeń mamy do czynienia również wspomnianą cechą oraz ceną ofertową. Współczynnik Giniego dla większości liści wynosi około 0,40. Sugeruje to stosunkowo duże zanieczyszczenie węzłów. Zarózno ze schematyu drzewa jak i z wykresu feature importance (rys. 23) można wyczytać, że najważniejszymi cechami decydującymi o klasyfikacji pod kątem rynku są cena za metr kwadratowy, odległość od centrum miasta oraz cena ofertowa.

Obraz zawierający diagram, tekst, linia

Opis wygenerowany automatycznie

Rysunek 22. Model drzew decyzyjnych - predykcja rynku

*Źródło:* Opracowanie własne

Obraz zawierający tekst, zrzut ekranu, diagram, Wykres

Opis wygenerowany automatycznie

Rysunek 23. Feature importance dla drzew decyzyjnych - predykcja rynku

*Źródło:* Opracowanie własne

# 6. Podsumowanie

Wyniki przedstawione w pracy miały na celu wykorzystanie web scrapingu do utworzenia wielkiego zbioru danych, a następnie na ich podstawie przetestowanie wybranych algorytmów uczenia maszynowego. Jak wspomniano, połączenie web scrapingu, technik zgłębiania danych oraz uczenia maszynowego może stanowić potężne narzędzie   
w przetwarzaniu i analizie danych z rynku nieruchomości. Jest to o tyle istotne, że takie rozwiązania są obecnie potrzebne dla wielu stron, których przedmiotem zainteresowania jest właśnie rynek nieruchomości. Jak wykazano w pracy wszystkie trzy elementy są ze sobą ściśle powiązane i razem mogą dawać oczekiwane oraz dokładne rezultaty.

Zastosowanie technik web scrapingu pozwoliło na skompletowanie zbioru danych liczącego 593 908 danych. Każde pobrane ogłoszenie zawierało 14 informacji tj. adres url, miasto, region, ulica, szerokość i długość geograficzna, liczba pokoi, piętro, powierzchnia, rok, waluta ceny, cena, rynek, nazwa oraz cena za metr kwadratowy. Ważnym aspektem danych pochodzących z web scrapingu są braki danych oraz wątpliwa jakość, które zostały wykryte na etapie eksploracyjnej analizy danych. Niektóre nieprawidłowo zapisane w ogłoszeniach informacje wpływały także na występowanie obserwacji odstających, co mogło zaburzyć późniejsze badania. Zbadane współzależności etapie eksploracyjnej analizy danych wykazały, że najistotniejszymi informacjami, które mogą mieć wpływ na zmienność cen jest powierzchnia mieszkania, liczba pokoi, cena za metr kwadratowy.

Poprzez testowanie wybranych metod uczenia maszynowego z wykorzystaniem zmiennych pochodzących z web scrapingu wykazano, że modele uczenia maszynowego mogą stanowić rozbudowane narzędzie w analizie danych z rynku nieruchomości. Można je wykorzystywać do wielu zagadnień związanych z tym rynkiem np. predykcja cen, badanie lokalnych rynków, zróżnicowanie przestrzenne, czy klasyfikacja.

Punktem wyjścia w modelowaniu było wykorzystanie regresji liniowej badającej współzależność najbardziej skorelowanych zmiennych tj. powierzchni mieszkań oraz ich cen. Model ten charakteryzował się jednak niską jakością oraz wysokimi wartościami błędów, co wynika zapewne z faktu, że na cenę mieszkania wpływa wiele zmiennych. Potwierdzeniem tego były wyniki modelu regresji wielomianowej, która często lepiej dopasowuje się do danych. Jednak w przypadku relacji dwóch wspomnianych zmiennych jakość modelu uległa bardzo nieznacznej poprawie.

W przypadku predykcji cen mieszkań, na które wpływa wiele czynników dobrym rozwiązaniem jest dodanie ich do modelu. Takie rozwiązanie można zbudować w oparciu o regresję wieloraką. Model z wieloma zmiennymi tj. liczba pokoi, piętro, powierzchnia, cena za metr kwadratowy, odległość od centrum okazał się być bardzo dobrej jakości. Nieocenionym aspektem tej techniki uczenia maszynowego jest uzyskanie konkretnych współczynników dla każdej zmiennej. Algorytm regresji wielorakiej może być jednak zbudowany w oparciu o zbyt wiele zmiennych, a taki model wykazywać cechy przeuczenia. Rozwiązaniem tego problemu może być regresja LASSO, w której nakłada się karę za zbytnią złożoność modelu. W tym przypadku obie techniki przyniosły jednak podobne rezultaty.

Dla tych samych zmiennych zbudowany został również algorytm drzew decyzyjnych oraz lasów losowych. W pierwszym przypadku model posiadał nieco mniejszą jakość niż model regresji wielorakiej i LASSO, a tym samym wyższe wartości błędów. Jednak algorytm pochodny, czyli lasy losowe wykazały w tym przypadku najlepsze zdolności predykcyjne. Jakość modelu można określić jako bardzo wysoką.

Bardzo zbliżone wartości w zakresie wyjaśniania zmienności cen mieszkań uzyskał algorytm k-NN. Jednak trzeba podkreślić, że w tym przypadku choć współczynnik determinacji wynosił prawie tyle samo co w przypadku algorytmu lasów losowych, to wartości błędów absolutnego, średniego błędu kwadratowego i jego pierwiastka były znacznie wyższe.

Rozszerzeniem modelowania z wykorzystaniem regresji może być dołączenie do modelu zmiennych kategorialnych. Dlatego też do wyżej wymienionych zmiennych numerycznych dodano cechy tj. miasto oraz rodzaj rynku. Jak można było oczekiwać dołączenie takich zmiennych poprawiło jakość modelu, a także spowodowało obniżenie błędów. W tym przypadku reszty regresji były praktycznie symetrycznie rozłożone względem zera, co sugeruje dobre zdolności predykcyjne.

Powszechnie stosowanymi algorytmami w zagadnieniach klasyfikacji są regresja logistyczna oraz drzewa decyzyjne. W przypadku regresji logistycznej zadaniem modelu było rozpoznanie na podstawie zmiennych numerycznych rozpoznanie czy mieszkanie zlokalizowane jest w Warszawie. W tym zakresie model lepiej rozpoznawał jednak mieszkanie nie mieszczące się w stolicy. Jakość algorytmu uległa znacznej poprawie po dodaniu do modelu zmiennej współrzędne geograficzne, co znacznie ułatwiło zadanie.

Ciekawym zagadnieniem w zakresie analizy danych z rynku nieruchomości jest także kategoryzacja czy mieszkanie pochodzi od dewelopera czy z rynku wtórnego. W tym przypadku porównanie modeli regresji logistycznej oraz drzew decyzyjnych pokazało, że lepiej z tym problemem radzi sobie ten drugi algorytm.

Bez wątpienia analiza rynku nieruchomości z wykorzystaniem web scrapingu   
i uczenia maszynowego może być rozszerzona o kolejne, interesujące zagadnienia oraz szukanie powiązań i zależności. Niniejsza praca udowadnia istnienie istotność wykorzystania web scrapingu jako źródła danych do późniejszej budowy modeli *machine learning*. Patrząc na główne zagadnienie, czyli predykcja cen mieszkań najlepszym rozwiązaniem okazuje się być wykorzystanie algorytmu lasów losowych, co jest zgodne z wynikami opisanych publikacji. Jednak trzeba podkreślić, że jakość modeli wzrasta wraz z liczbą zmiennych. Wynika to ze złożoności nieruchomości, na których cenę wpływ ma wiele czynników i cech.

# Spis literatury

Banerjee D., Dutta S. (20107), Predicting the housing price direction using machine learning techniques, IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), s. 2998-3000

Bartłomiejczyk A., Ptach D., Wata M. (2020), Elementy uczenia maszynowego na zajęciach matematyki, Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki PG, 71, s. 27-30

Bini S.A., (2018). Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care?, The Journal of Arthroplasty, 33(8), s.2358–2361

Bolc L., Zaremba J., (1992), Wprowadzenie do uczenia się maszyn (Tom 1), Akademicka Oficyna Wydawnicza RM.

Breiman L., (2001), Random forests [w:] Król-Nowak A., Kotarba K., (2022), Podstawy uczenia maszynowego, Oficyna Wydawnictwa AGH, Kraków

Bruce P., Bruce A., Gedeck P., (2021), Statystyka praktyczna w Data Science – 50 kluczowych zagadnień w językach R i Python, Helion, Gliwice

Ceh M., Kilibarda M., Lisec A., Bajat B. (2018), Estimating the Performance of Random Forest versus Multiple Regression for Predicting Prices of the Apartments, ISPRS International Journal of geo-information, 7(5), 168

Cichosz P., (2003), Systemy uczące się, PWN, Warszawa

Cutler A., Cutler D.R., Stevens J.R., (2012), *Random Forests* [w:] Zhang C., Ma Y., Ensemble Machine Learning, Springer, New York, s. 157-175

Dhar T., Manikandan P., (2023), A Literature Review on Using Machine Learning Algorithm to Predict House Prices, International Conference on Intelligent Computing Technologies and Research, s. 132-137

Fan C., Cui Z., Zhong X., (2018), House Prices Prediction with Machine Learning Algorithms, Proceedings of the 2018 10th International Conference on Machine Learning and Computing – ICMLC

Flach P., (2012), Machine Learning: The Art and Science of Algorithms, s. 3

Geron A., (2023), Uczenie maszynowe z użyciem Scikit-Learn, Keras i TensorFlow, Helion, Gliwice

Ghosalkar N.N., Dhage S.N., (2018), Real Estate Value Prediction Using Linear Regression, Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)

Grybauskas A., Pilinkienė V., Stundžienė A., (2021), Predictive analytics using Big Data for the real estate market during the COVID‑19 pandemic, Journal of Big Data, 8(1), s. 1-20

Hong J, .(2020), An Application of XGBoost, LightGBM, CatBoost Algorithms on House Price Appraisal System, Housing Finance Research, 4, s. 33–64

Hong J., Choi H., Kim W.S., (2020), A house price valuation based on the random forest approach: The mass appraisal of residential property in South Korea, International Journal of Strategic Property Management, 24, s. 140–152

Hu L., He S., Han Z., Xiao H., Su S., Weng M., Cai Z., (2019), Monitoring housing rental prices based on social media: An integrated approach of machine-learning algorithms and hedonic modeling to inform equitable housing policies, Land Use Policy, 82, s. 657–673

Ja’afar N. S., Mohamad J., Ismail S. (2021), Machine learning for property price prediction and price valuation: a systematic literature review, Planning Malaysia, 19

Khan H. A., Rehman J., (2023), Applying Machine Learning Models for Forecasting House Prices - A Case of the Metropolitan City of Karachi, Journal of Entrepreneurship, Management and Innovation, 5(3), s. 376-400

Khare S., Gourisaria M.K., Harshvardhan G.M., Joardar S., Singh V., (2021), Real estate cost estimation through data mining techniques, IOP Conference Series: Materials Science and Engineering, 1099(1)

Kok N., Koponen E.L., Martínez-Barbosa C.A., (2017), Big Data in Real Estate? From Manual Appraisal to Automated Valuation, The Journal of Portfolio Management, 43(6), s. 202–211

Korczak K., (2019), Uczenie maszynowe w opiece zdrowotnej, Roczniki Kolegium Analiz Ekonomicznych SGH, s. 305-316

Kuvalekar A., Manchewar S., Mahadik S., Jawale S., (2020), House Price Forecasting Using Machine Learning, Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST)

Lasota-Kapczuk A., (2021), Sztuczna inteligencja w kształtowaniu bezpieczeństwa państwa w II dekadzie XXI wieku, Instytut Nauk o Bezpieczeństwie, Uniwersytet w Siedlcach

Liu Z., (2023), Real Estate Price Prediction based on Supervised Machine Learning Scenarios. Highlights in Science, Engineering and Technology, 39, s. 731-737

Lu S., Li Z., Qin Z., Yang X., Goh R.S.M., (2017), A hybrid regression technique for house prices prediction, IEEE international conference on industrial engineering and engineering management (IEEM), s. 319-323

Manjula R., Jain S., Srivastava S., Kher P.R., (2017), Real estate value prediction using multivariate regression models, IOP Conferenece Series: Materials Science and Engineering, 263(4)

Michalski R.S., Carbonell J.G., Mitchell T.M., (1983), Machine Learning - an artificial intelligence approach, (2), s. 3-26

Mitchell T. M., (1997), Machine Learning, McGraw-Hill Science

Mu J., Wu F., Zhang A., (2014), Housing Value Forecasting Based on Machine Learning Methods, Abstract and Applied Analysis, s. 1-7

Neloy A.A., Haque H.M., Ul Islam M.M., (2019), Ensemble Learning Based Rental Apartment Price Prediction Model by Categorical Features Factoring, In Proceedings of the 2019 11th International Conference on Machine Learning and Computing, s. 350–356

Oluwunmi A.O., Role B.A., Akinwale O.M., Oladayo O.P., Afolabi T.O., (2019), Big Data and real estate: a review of literature, International Conference on Engineering for Sustainable World, s. 1-7

Park B., Bae J.K, (2015), Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data, Expert Systems with Applications, s. 2928–2934

Patel J.M., (2020), Getting Structured Data from the Internet: Running Web Crawlers/Scrapers on a Big Data Production Scale, Apress

Phan T.D., (2018) Housing price prediction using machine learning algorithms: the case of Melbourne city, Australia, 2018 International conference on machine learning and data engineering (iCMLDE), s. 35-42

Pow N., Janulewicz E., Liu L., (2014), Applied Machine Learning Project 4 Prediction of real estate property prices in Montreal, Course project, COMP-598, McGill University

Putri M.R, Wijaya I.G., Praja F.P., Hadi A., Hamami F., (2023), The Comparison Study of Regression Models (Multiple Linear Regression, Ridge, Lasso, Random Forest, and Polynomial Regression) for House Price Prediction in West Nusa Tenggara, 2023 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS), s. 1-6

Raschka S., Mirjalili V., (2017), Python Uczenie maszynowe, Helion, Gliwice

Sampathkumar V., Santhi M. H., Vanjinathan, J., (2015), Forecasting the land price using statistical and neural network software, Procedia Computer Science, 57, s. 112-121

Samuel A.L., (1959), Some Studies in Machine Learning Using the Game of Checkers, IBM Journal of research and development, 3(3), s. 210-229

Szeliga M., (2017), Data Science i uczenie maszynowe, PWN, Warszawa

Thamarai M., Malarvizhi S.P., (2020) House price prediction modeling using machine learning. Internal Journal of Information Engineering & Electronic Business, 12(2)

Truong Q., Nguyen M., Dang H., Mei B., (2020), Housing Price Prediction via Improved Machine Learning Techniques, 2019 International Conference on Identification, Information and Knowledge in the Internet of Things

Tsonev A., Dyankova V., Yusufov, Y., (2022). Using web scraping for real estate price analysis, Mathematical and Software Engineering, 8(1-2), s. 19-23

Uzun E., Novel A., (2020), Web Scraping Approach Using the Additional Information Obtained From Web Pages, IEEE Access, 8

Weiss S.M., Kulikowski C. A., (1991), Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems, Morgan Kaufmann Publishers Inc

Winson-Geideman K., Krause A., (2016), Transformations in real estate research: The big data revolution, In Proceedings of the 22nd annual pacific-rim real estate society conference, s. 17-20

Varma A., Sarma A., Doshi S., Nair R, (2018), House Price Prediction Using Machine Learning and Neural Networks, 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), s. 1936-1939

Voutas Chatzidis I., (2019), Prediction of Housing Prices based on Spatial & Social Parameters using Regression & Deep Learning Methods, University of Thessaloniki, Thessaloniki

Yu H., Wu J., (2016), Real Estate Price Prediction with Regression and Classification, CS 229 Autumn 2016 Project Final Report s. 1–5

Zając S., (2022), Modelowanie dla biznesu. Analityka w czasie rzeczywistym – narzędzia informatyczne i biznesowe, Oficyna Wydawnicza SGH, Warszawa

Zych M., (2021), Web scraping w analityce danych iw badaniach mixed-method Research, Instytut Zarządzania Informacją Uniwersytet Jagielloński

# Spis rycin

[Rysunek 1. Schemat uczenia maszynowego 11](#_Toc156755359)

[Rysunek 2. Zmienne oraz ich typy pobrane w wyniku web scraping 31](#_Toc156755360)

[Rysunek 3. Histogramy oraz wykresy pudełkowe przed usunięciem outlierów 33](#_Toc156755361)

[Rysunek 4. Histogramy oraz wykresy pudełkowe po usunięciu outlierów 36](#_Toc156755362)

[Rysunek 5. Macierz korelacji dla zmiennych numerycznych 37](#_Toc156755363)

[Rysunek 6. Średnie ceny ofertowe mieszkań w poszczególnych miastach 38](#_Toc156755364)

[Rysunek 7. Średnie ceny ofertowe w podziale na rodzaj rynku 39](#_Toc156755365)

[Rysunek 8. Model regresji liniowej 40](#_Toc156755366)

[Rysunek 9. Reszty regresji liniowej 41](#_Toc156755367)

[Rysunek 10. Stopień wielomianu dla regresji wielomianowej w porównaniu do błędu MSE 42](#_Toc156755368)

[Rysunek 11. Model regresji wielomianowej ze stopniem wielomianu równym 6 42](#_Toc156755369)

[Rysunek 12. Reszty regresji wielorakiej 44](#_Toc156755370)

[Rysunek 13. Model regresji ze zmiennymi kategorialnymi 45](#_Toc156755371)

[Rysunek 14. Histogram reszt regresji ze zmiennymi kategorialnymi 45](#_Toc156755372)

[Rysunek 15. Reszty regresji ze zmiennymi kategorialnymi w podziale na miasta 46](#_Toc156755373)

[Rysunek 16. Reszty regresji ze zmiennymi kategorialnymi w podziale na rodzaj rynku 47](#_Toc156755374)

[Rysunek 17. Model drzew decyzyjnych 49](#_Toc156755375)

[Rysunek 18. Model lasów losowych 50](#_Toc156755376)

[Rysunek 19. Liczba sąsiadów algorytmu k-NN w porównaniu z błędem RMSE 51](#_Toc156755377)

[Rysunek 20. Model k-NN 51](#_Toc156755378)

[Rysunek 21. Krzywa ROC dla dwóch modeli regresji logistycznej 53](#_Toc156755379)

[Rysunek 22. Model drzew decyzyjnych - predykcja rynku 55](#_Toc156755380)

[Rysunek 23. Feature importance dla drzew decyzyjnych - predykcja rynku 55](#_Toc156755381)

# Spis tabel

[Tabela 1. Liczba pobranych i dostępnych ogłoszeń mieszkań 23](#_Toc156738983)

[Tabela 2. Macierz pomyłek 30](#_Toc156738984)

[Tabela 3. Statystyki opisowe poszczególnych zmiennych przed usunięciem outlierów 32](#_Toc156738985)

[Tabela 4. Statystyki opisowe zmiennych po usunięciu obserwacji odstających 34](#_Toc156738986)

[Tabela 5. Współczynniki regresji wielorakiej dla poszczególnych zmiennych 43](#_Toc156738987)

[Tabela 6. Parametry i oceny jakości dla modelu drzew decyzyjnych 48](#_Toc156738988)

[Tabela 7. Macierz pomyłek dla modelu regresji logistycznej 52](#_Toc156738989)

[Tabela 8. Macierz pomyłek dla modelu regresji logistycznej z dodatkowymi zmiennymi 52](#_Toc156738990)

[Tabela 9. Macierz pomyłek dla modelu regresji logistycznej klasyfikujący rodzaj rynku 54](#_Toc156738991)

[Tabela 10. Macierz pomyłek dla drzewa decyzyjnego - predykcja rynku 54](#_Toc156738992)

# Załączniki

# Scrapper

def get\_offers(page\_number):

url = f'https://gratka.pl/nieruchomosci/mieszkania/warszawa/wtorny?page={page\_number}&cena-calkowita:min=1'

headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_10\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36'}

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

sleep(randint(60, 80))

content = response.text

soup = BeautifulSoup(content, 'html.parser')

script\_tag = soup.find('script', {'type': "application/ld+json"})

if script\_tag:

return script\_tag.get\_text()

return None

def process\_page(json\_text):

if json\_text is None:

return None

data = json.loads(json\_text)

urls = []

localities = []

regions = []

street\_addresses = []

latitudes = []

longitudes = []

num\_rooms = []

floor\_levels = []

floor\_sizes = []

year\_builts = []

currencies = []

prices = []

apartment\_names = []

for offer in data["mainEntity"]["itemListElement"]:

item\_offered = offer["itemOffered"]

urls.append(item\_offered.get("url", None))

localities.append(item\_offered["address"].get("addressLocality", None))

regions.append(item\_offered["address"].get("addressRegion", None))

street\_addresses.append(item\_offered["address"].get("streetAddress", None))

if "geo" in item\_offered:

latitudes.append(item\_offered["geo"].get("latitude", None))

longitudes.append(item\_offered["geo"].get("longitude", None))

else:

latitudes.append(None)

longitudes.append(None)

num\_rooms.append(item\_offered.get("numberOfRooms", None))

floor\_levels.append(item\_offered.get("floorLevel", None))

floor\_sizes.append(item\_offered["floorSize"].get("value", None))

year\_builts.append(item\_offered.get("yearBuilt", None))

currencies.append(offer.get("priceCurrency", None))

prices.append(offer.get("price", None))

if "name" in item\_offered:

apartment\_names.append(item\_offered["name"])

else:

apartment\_names.append(None)

df = pd.DataFrame({

"URL": urls,

"Address Locality": localities,

"Address Region": regions,

"Street Address": street\_addresses,

"Latitude": latitudes,

"Longitude": longitudes,

"Number of Rooms": num\_rooms,

"Floor Level": floor\_levels,

"Floor Size": floor\_sizes,

"Year Built": year\_builts,

"Price Currency": currencies,

"Price": prices,

"Rynek": "wtorny",

"Apartment Name": apartment\_names

})

return df

**# Model regresji liniowej**

X = calosc\_model[['Powierzchnia']]

y = calosc\_model['Cena']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

lr = LinearRegression()

lr.fit(X\_train, y\_train)

y\_pred = lr.predict(X\_test)

plt.figure(figsize=(12, 8))

plt.scatter(X\_test['Powierzchnia'], y\_test, color='lightsteelblue', label='Actual Prices')

plt.plot(X\_test['Powierzchnia'], y\_pred, color='red', linewidth=3, label='Model regresji')

plt.xlabel('Powierzchnia', fontsize=12, fontname="Times New Roman")

plt.ylabel('Cena', fontsize=12, fontname="Times New Roman")

#plt.title('Regresja liniowa: Cena vs. Powierzchnia', fontsize=14)

#plt.legend()

plt.savefig('regresja.jpg', bbox\_inches='tight', dpi=1200)

plt.show()

**# Model regresji wielomianowej**

X = calosc\_model[['Powierzchnia']]

y = calosc\_model['Cena']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

max\_degree = 15

degrees = np.arange(1, max\_degree + 1)

mse\_values = []

for degree in degrees:

poly = PolynomialFeatures(degree=degree)

X\_train\_poly = poly.fit\_transform(X\_train)

X\_test\_poly = poly.transform(X\_test)

poly\_reg = LinearRegression()

poly\_reg.fit(X\_train\_poly, y\_train)

y\_pred\_poly = poly\_reg.predict(X\_test\_poly)

mse = mean\_squared\_error(y\_test, y\_pred\_poly)

mse\_values.append(mse)

plt.figure(figsize=(10, 6))

plt.plot(degrees, mse\_values, marker='o', linestyle='-', color='b')

plt.xlabel('Stopień wielomianu', fontsize=12, fontname="Times New Roman")

plt.ylabel('Błąd MSE', fontsize=12, fontname="Times New Roman")

#plt.title('Zależność Błędu MSE od Stopnia Wielomianu', fontsize=14)

plt.xticks(degrees)

plt.grid(True)

plt.savefig('stopien wielomianu.jpg', bbox\_inches='tight', dpi=1200)

plt.show()

**# Model regresji wielorakiej**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = math.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

print('MAE:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print(f'Mean Squared Error (MSE): {mse}')

print(f'Root Mean Squared Error (RMSE): {rmse}')

print(f'R-squared: {r2}')

coefficients = model.coef\_

intercept = model.intercept\_

print(f'Współczynniki regresji: {coefficients}')

print(f'Przesunięcie (intercept): {intercept}')

**# Model regresji LASSO**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

lr = LinearRegression()

lr.fit(X\_train, y\_train)

y\_pred\_lr = lr.predict(X\_test)

lasso = Lasso(alpha=100) # Parametr alpha kontroluje siłę regularyzacji, możesz dostosować jego wartość

lasso.fit(X\_train, y\_train)

y\_pred\_lasso = lasso.predict(X\_test)

ridge = Ridge(alpha=1.0) # Podobnie, parametr alpha reguluje siłę regularyzacji

ridge.fit(X\_train, y\_train)

y\_pred\_ridge = ridge.predict(X\_test)

mse\_lr = mean\_squared\_error(y\_test, y\_pred\_lr)

mse\_lasso = mean\_squared\_error(y\_test, y\_pred\_lasso)

mse\_ridge = mean\_squared\_error(y\_test, y\_pred\_ridge)

rmse\_lr = math.sqrt(mse\_lr)

rmse\_lasso = math.sqrt(mse\_lasso)

rmse\_ridge = math.sqrt(mse\_ridge)

mae\_lasso = metrics.mean\_absolute\_error(y\_test, y\_pred\_lasso)

mae\_ridge = metrics.mean\_absolute\_error(y\_test, y\_pred\_ridge)

r2\_lasso = r2\_score(y\_test, y\_pred\_lasso)

r2\_ridge = r2\_score(y\_test, y\_pred\_ridge)

print(f'RMSE dla regresji liniowej: {rmse\_lr}')

print(f'RMSE dla regresji z regularyzacją L1 (Lasso): {rmse\_lasso}')

print(f'RMSE dla regresji z regularyzacją L2 (Ridge): {rmse\_ridge}')

print(f'MAE dla regresji z regularyzacją L1 (Lasso): {mae\_lasso}')

print(f'MAE dla regresji z regularyzacją L2 (Ridge): {mae\_ridge}')

print(f'R^2 dla regresji z regularyzacją L1 (Lasso): {r2\_lasso}')

print(f'R^ dla regresji z regularyzacją L2 (Ridge): {r2\_ridge}')

plt.figure(figsize=(12, 8))

plt.scatter(X\_test['Powierzchnia'], y\_test, color='lightsteelblue', label='Actual Prices')

plt.plot(X\_test['Powierzchnia'], y\_pred\_lr, color='red', linewidth=3, label='Regresja liniowa')

plt.plot(X\_test['Powierzchnia'], y\_pred\_lasso, color='green', linewidth=3, label='Lasso')

plt.plot(X\_test['Powierzchnia'], y\_pred\_ridge, color='blue', linewidth=3, label='Ridge')

plt.xlabel('Powierzchnia', fontsize=12, fontname="Times New Roman")

plt.ylabel('Cena', fontsize=12, fontname="Times New Roman")

plt.title('Porównanie regresji liniowej, Lasso i Ridge', fontsize=14)

plt.legend()

plt.savefig('regresja\_porownanie.jpg', bbox\_inches='tight', dpi=1200)

plt.show()

**# Model regresji ze zmiennymi kategorialnymi**

features = ['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum', 'Miasto', 'Rynek']

X = calosc\_model[features]

y = calosc\_model['Cena']

categorical\_cols = ['Miasto', 'Rynek']

preprocessor = ColumnTransformer(

transformers=[

('num', SimpleImputer(strategy='mean'), ['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum']),

('cat', OneHotEncoder(), categorical\_cols)

])

model = Pipeline(steps=[('preprocessor', preprocessor),

('regressor', LinearRegression())])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = math.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

print('MAE:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print(f'MSE: {mse}')

print(f'RMSE: {rmse}')

print(f'R^2: {r2}')

**# Model drzew decyzyjnych**

X = calosc\_model[['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum']]

y = calosc\_model['Cena']

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='mean') wypełniania

X = imputer.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

tree\_model = DecisionTreeRegressor(max\_depth=7, min\_samples\_leaf=1, min\_samples\_split=2, random\_state=42)

tree\_model.fit(X\_train, y\_train)

y\_pred = tree\_model.predict(X\_test)

mae = mean\_absolute\_error(y\_test, y\_pred)

print("Mean Absolute Error (MAE):", mae)

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error (MSE):", mse)

rmse = np.sqrt(mse)

print("Root Mean Squared Error (RMSE):", rmse)

r2 = r2\_score(y\_test, y\_pred)

print("R2:", r2)

**# Model lasów losowych**

num\_features = ['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw','odleglosc\_od\_centrum']

numeric\_transformer = Pipeline(steps=[

('imputer', SimpleImputer(strategy='mean'))

])

preprocessor = ColumnTransformer(

transformers=[

('num', numeric\_transformer, num\_features)

])

rf\_model = Pipeline(steps=[

('preprocessor', preprocessor),

('regressor', RandomForestRegressor(random\_state=42))

])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

rf\_model.fit(X\_train, y\_train)

y\_pred\_rf = rf\_model.predict(X\_test)

mse\_rf = mean\_squared\_error(y\_test, y\_pred\_rf)

mae\_rf = mean\_absolute\_error(y\_test, y\_pred\_rf)

rmse\_rf = math.sqrt(mse\_rf)

r2\_rf = r2\_score(y\_test, y\_pred\_rf)

print(f'Mean Absolute Error (MAE) dla Random Forest: {mae\_rf}')

print(f'Mean Squared Error (MSE) dla Random Forest: {mse\_rf}')

print(f'Root Mean Squared Error (RMSE) dla Random Forest: {rmse\_rf}')

print(f'R-squared dla Random Forest: {r2\_rf}')

**# Model k-NN**

X = calosc\_model[['Liczba pokoi', 'Piętro','Powierzchnia','Cena\_mkw','odleglosc\_od\_centrum']]

y = calosc\_model['Cena']

imputer = SimpleImputer(strategy='mean') # dostosuj strategię wypełniania

X = imputer.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

knn\_model = KNeighborsRegressor(n\_neighbors=14) # dostosuj liczbę sąsiadów

knn\_model.fit(X\_train\_scaled, y\_train)

y\_pred = knn\_model.predict(X\_test\_scaled)

mae = mean\_absolute\_error(y\_test, y\_pred)

print("Mean Absolute Error (MAE):", mae)

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error (MSE):", mse)

rmse = np.sqrt(mse)

print("Root Mean Squared Error (RMSE):", rmse)

r2 = r2\_score(y\_test, y\_pred)

print("R-squared (R2):", r2)

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred)

plt.plot([min(y\_test), max(y\_test)], [min(y\_test), max(y\_test)], linestyle='--', color='red', linewidth=2)

#plt.title('Predykcje vs. Prawdziwe wartości dla k-NN')

plt.xlabel('Prawdziwe wartości',fontsize=12, fontname="Times New Roman")

plt.ylabel('Predykcje',fontsize=12, fontname="Times New Roman")

plt.savefig('knn.jpg', bbox\_inches='tight', dpi=1200)

plt.show()

**# Model regresji logistycznej**

X = ['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum']

X1 = calosc\_model[['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum']]

X = ['Długość geo', 'Szerokość geo', 'Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum']

X2 = calosc\_model[['Długość geo', 'Szerokość geo', 'Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum']]

def build\_and\_evaluate\_model(X, y):

imputer = SimpleImputer(strategy='mean')

X = imputer.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

model = LogisticRegression(random\_state=42)

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

tn, fp, fn, tp = conf\_matrix.ravel()

sensitivity = tp / (tp + fn)

specificity = tn / (tn + fp)

report = classification\_report(y\_test, y\_pred, target\_names=['Not Warsaw', 'Warsaw'])

fpr, tpr, thresholds = roc\_curve(y\_test, model.predict\_proba(X\_test\_scaled)[:, 1])

roc\_auc = auc(fpr, tpr)

return accuracy, sensitivity, specificity, report, fpr, tpr, roc\_auc

y1 = np.where(calosc\_model['Miasto'] == 'Warszawa', 1, 0)

accuracy1, sensitivity1, specificity1, report1, fpr1, tpr1, roc\_auc1 = build\_and\_evaluate\_model(X1, y1)

y2 = np.where(calosc\_model['Miasto'] == 'Warszawa', 1, 0)

accuracy2, sensitivity2, specificity2, report2, fpr2, tpr2, roc\_auc2 = build\_and\_evaluate\_model(X2, y2)

plt.figure(figsize=(10, 6))

plt.plot(fpr1, tpr1, color='darkorange', lw=2, label=f'Model 1 (AUC = {roc\_auc1:.2f})')

plt.plot(fpr2, tpr2, color='green', lw=2, label=f'Model 2 (AUC = {roc\_auc2:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Losowy klasyfikator')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate', fontsize=12, fontname="Times New Roman")

plt.ylabel('True Positive Rate', fontsize=12, fontname="Times New Roman")

plt.title('Krzywe ROC dla dwóch modeli', fontsize=14, fontname="Times New Roman")

plt.legend(loc="lower right")

plt.savefig('roc\_models.jpg', bbox\_inches='tight', dpi=1200)

plt.show()

print('Model 1:')

print(f'Accuracy: {accuracy1}')

print(f'Sensitivity: {sensitivity1}')

print(f'Specificity: {specificity1}')

print('Classification Report:\n', report1)

print('\nModel 2:')

print(f'Accuracy: {accuracy2}')

print(f'Sensitivity: {sensitivity2}')

print(f'Specificity: {specificity2}')

print('Classification Report:\n', report2)

**# Model drzew decyzyjnych – klasyfikacja**

X = calosc\_model[['Liczba pokoi', 'Piętro', 'Powierzchnia', 'Cena\_mkw', 'odleglosc\_od\_centrum', 'Cena']]

y = np.where(calosc\_model['Rynek'] == 'pierwotny', 1, 0)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

tree\_model = DecisionTreeClassifier(max\_depth=3, min\_samples\_leaf=1, min\_samples\_split=2, random\_state=42)

tree\_model.fit(X\_train, y\_train)

y\_pred = tree\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

print(classification\_report(y\_test, y\_pred)

# Streszczenie

W pracy przedstawiono metody web scrapingu oraz uczenia maszynowego, które mogą posłużyć do zbierania i analizy danych z rynku nieruchomości. Materiałem źródłowym był zbiór danych jako efekt narzędzia pobierającego informacje o ogłoszeniach mieszkań   
z serwisu nieruchomościowego. Zbudowana baza danych liczyła 42 422 rekordy oraz 14 kolumn (cech). Zbiór ten został poddany eksploracyjnej analizie danych, gdzie m.in. usunięto obserwacje odstające i dokonano analizy statystycznej zmiennych. W oparciu o tak przygotowaną bazę dane posłużyły do budowy modeli *machine learning* tj. regresje liniowa, wielomianowa, wieloraka, ze zmiennymi kategorialnymi, logistyczna, drzewa decyzyjne, lasy losowe oraz k-NN.

Wyniki wskazują, że najlepszym modelem do predykcji cen mieszkań na podstawie zmiennych numerycznych (liczba pokoi, piętro, powierzchnia, cena za metr kwadratowy, odległość od centrum) był algorytm lasów losowych. W przypadku zagadnień klasyfikacyjnych nieznacznie lepsze wyniki predykcyjne uzyskały drzewa decyzyjne.