Przewidywanie cen mieszkań na podstawie danych z ogłoszeń z serwisu Otodom

Jakub Stępkowski (272896)
Politechnika Wrocławska

Czerwiec 2024

1 Wstęp

Analiza rynku nieruchomości i ich wycena to kluczowe aspekty zarówno dla inwestorów, jak i dla indywidualnych nabywców. W dobie rosnącej ilości danych oraz zaawansowanych technologii, modele uczenia maszynowego stają się coraz bardziej popularnym narzędziem do prognozowania wartości nieruchomości. Uczenie maszynowe, umożliwia analizę dużych zbiorów złożonych danych i odkrywanie wzorców, które mogą być trudne do zauważenia przez człowieka.

W poniższej pracy, na podstawie zebranych danych z ogłoszeń nieruchomości, zostanie przeprowadzona analiza oraz trening modeli uczenia maszynowego, których celem będzie oszacowanie wartości rynkowej nieruchomości.

2 Pozyskanie zbioru danych

Pierwszym krokiem projektu było zgromadzenia jakościowego i rozbudowanego zbioru danych o mieszkaniach. Wybór padł na portal z ogłoszeniami *otodom.pl*, praca została skupiona na ogłoszeniach mieszkań na sprzedaż we Wrocławiu.

W celu zebrania danych, został zaimplementowany scraper, będący częścią większego projektu oprogramowania przetwarzającego ogłoszenia o nieruchomościach, które cały czas jest rozwijane.

Proces scrapowania ogłoszeń dzieli się na 2 części. Pierwsza z nich polega na pobieraniu podstawowych danych o nieruchomości z widoku listy ogłoszeń. Pobierane są dane takie jak tytuł, adres, cena, piętro, powierzchnia oraz link do podstrony, który jest kluczowy w drugim etapie. Etap drugi polega na odwiedzaniu podstron kolejnych mieszkań w celu scrapingu bardziej szczegółowych danych takich jak wysokość budynku (maksymalne piętro), wysokość czynszu, certyfikat energetyczny, forma własności, stan wykończenia, informację o posiadaniu balkonu, ogrodu lub tarasu, miejsca parkingowego, rodzaj ogrzewania, opis ogłoszenia, rynek (wtórny, pierwotny), typ ogłoszeniodawcy, rok budowy, typ budynku, informacje o oknach i windzie.

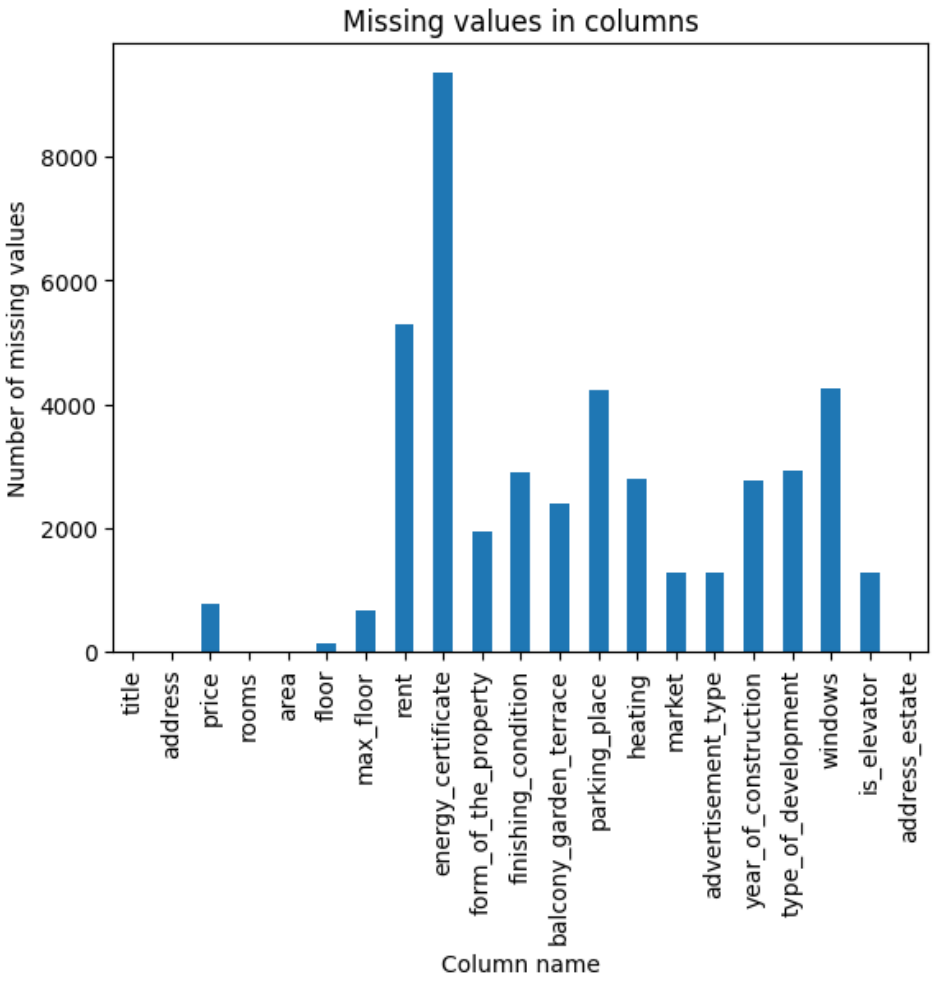
Implementacja została oparta na frameworku Pythona - Djnago, względu na szerokie plany rozbudowy, oraz ORM'a (Object–Relational Mapping) i panel administratora, który ten framework oferuje. Sam w sobie scraping odbywa się za pomocą modułu *requests* i *lxml* ze względu na możliwośc odwowyłania się do komponentów za pomocą xpath'a. Dane zapisywane są w bazie danych, ostatnim krokiem jest eksport wybranych atrybutów do pliku CSV.

3 Przetwarzanie wstępne

Po wyeksportowaniu zbiór danych zawiera następujące cechy:

1. title (tekst) - Tytuł ogłoszenia
2. address (tekst) - Adres mieszkania. W formacie ulica osiedle, dzielnica, miasto województwo, gdzie bardziej szczegółowe parametry mogą być pominięte.
3. price (liczba) - Cena mieszkania. Nie uwzględniamy ofert bez ceny mimo tego, że takie również występują.
4. rooms (liczba) - Liczba pokoi w mieszkaniu
5. area (liczba) - Powierzchnia w m^2
6. floor (tekst) - Piętro na którym znajduje się mieszkanie. Przy piętrach 10+ tracimy dokładną informacje. Występują rownież wartości takiej jak "parter", "suterena", "poddasze".
7. max_floor (liczba) - Liczba pięter w budynku
8. rent (liczba) - Wysokość czynszu
9. energy_certificate (tekst) - Certyfikat energetyczny
10. form_of_the_property - Forma własności
11. finishing_condition - Stan wykończenia
12. balcony_garden_terrace - Informację o posiadaniu balkonu, ogrodu lub tarasu
13. parking_place (tekstowe) - Miejsce parkingowe
14. heating (tekstowe) - Rodzaj ogrzewania
15. market (logiczne) - Rynek (wtórny bądź pierwotny)
16. advertisement_type (logiczne) - Typ ogłoszeniodawcy
17. year_of_construction (liczbowe) - Rok budowy
18. type_of_development (enumeryczne) - Typ budynku
19. finishing_condition (enumeryczne) - Stan wykończenia
20. windows (enumetyczne) - Informacja o materiale okien
21. is_elevator (logiczne) - Czy jest winda

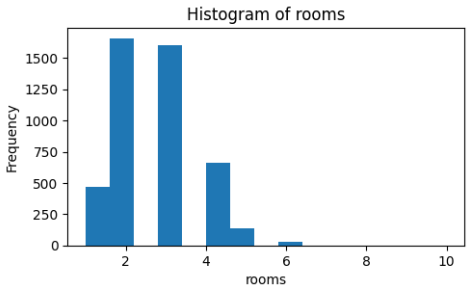
Z adresu została wykorzystania wyłącznie nazwa osiedla, na którym znajduje się mieszkanie. Umieszczono ją w kolumnie *adress_estate*. Cały zbiór zawiera 9843 mieszkań, jednak nie wszystkie oferty mają uzupełnione wszystkie cechy. Poniższy wykres pokazuje, dla każdej cechy, w ilu próbkach jest ona pusta:

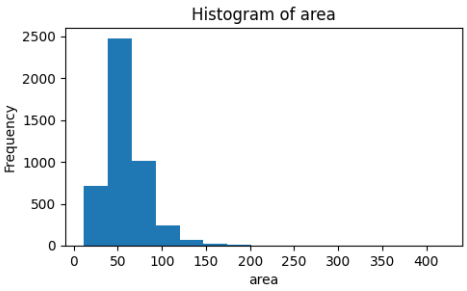


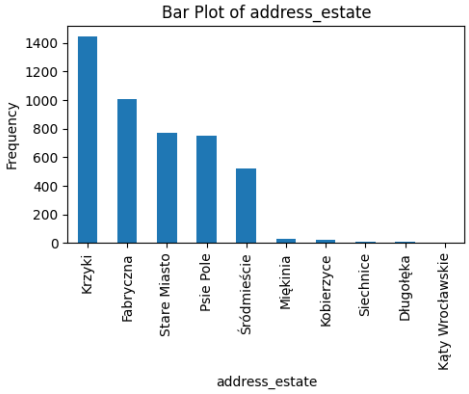
Po analizie brakujących danych postanowiono wykorzystać następujące cechy: rooms, area, address_estate, floor, max_floor, market, year_of_construction, finishing_condition, is_elevator, biorąc tylko te elementy, które posiadają wartość wszystkich atrybutów.

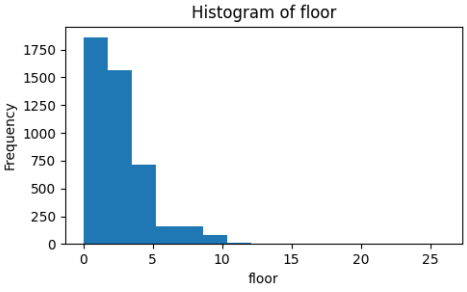
Zostały dokowane również poprawki szczególnych wartości w danych takie jak zamiana słownego "parter" na 0 dla *floor* (piętra) oraz ustaliliśmy jako połowę wartości max_floor dla przypadków z wartościami "10+".

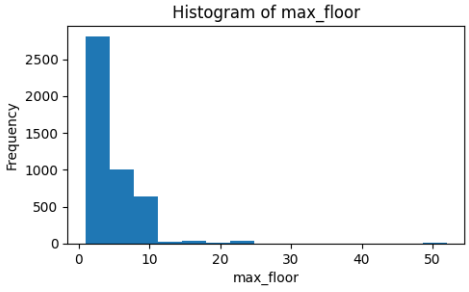
Poniższy wykres to histogram przestawiający rozkład wartości dla poszczególnych atrybutów:

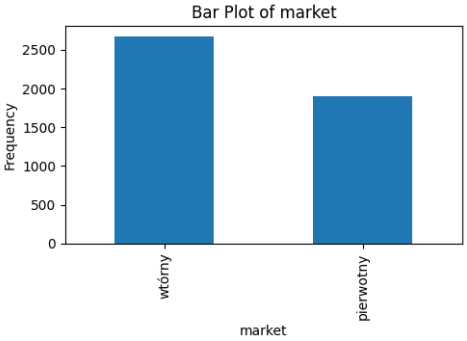

Histogram of rooms
Frequency
rooms

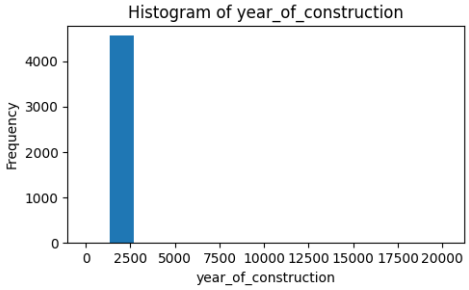

Histogram of area
Frequency
area

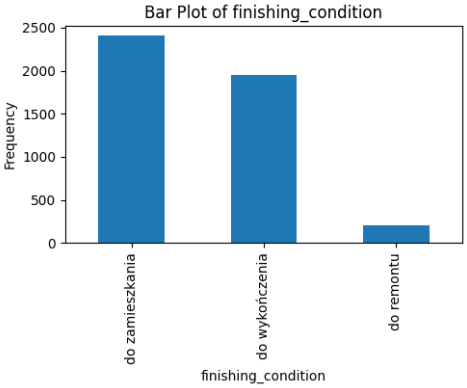

Bar Plot of address_estate
Frequency
Krzyki
Fabryczna
Stare Miasto
Psie Pole
Śródmieście
Miękinia
Kobierzyce
Siechnice
Długołęka
Kąty Wrocławskie
address_estate

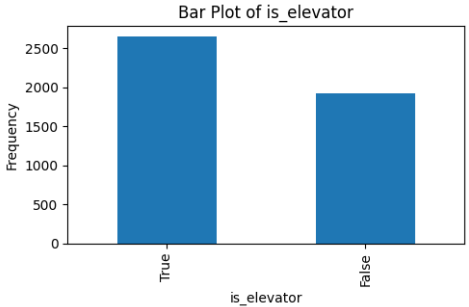

Histogram of floor
Frequency
floor

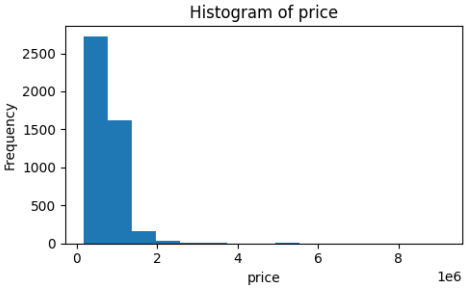

Histogram of max_floor
Frequency
max_floor


Bar Plot of market
Frequency
wtórny
pierwotny
market


Histogram of year_of_construction
Frequency
year_of_construction


Bar Plot of finishing_condition
Frequency
do zamieszkania
do wykończenia
do remontu
finishing_condition


Bar Plot of is_elevator
Frequency
True
False
is_elevator


Histogram of price
Frequency
price
1e6

Wykres ten umożliwił lokalizację i na bazie niego zostały wykluczone rekordy ze skrajnymi przypadkami wartości wynikającymi z pomyłki lub szczególnych przypadków takich jak rok budowy "6", "20244" czy też mieszkań droższych niż 2 miliony złotych.

Wypiszmy dokładnie ilość mieszkań dla konkretnych osiedli:

Address Estate	**Amount**
Krzyki	1449
Fabryczna	1008
Stare Miasto	769
Psie Pole	750
Śródmieście	523
Miękinia	32
Kobierzyce	22
Siechnice	7
Długołęka	6
Kąty Wrocławskie	1

Tabela 1: Tabele ilości mieszkań na osiedlach

Dostrzegamy, że należy również usunąć ofertę z Kątów Wrocławskich ponieważ pojedyńcze mieszkanie z tego regionu może zaburzać wynik, co dzieje się w szczególnosci w tym przypadku z uwagi na cene mieszkania znacząco przewyższającą średnią.

Po następujących operacjach, zbiór danych został zredukowany do 4496 elementów i wygląda w sposób następujący:

- Zawiera cechy kategorialne adress_estate, market i finishing_condition, is_elevator.
- Cechy binarne (market i is_elevator) zakodowano przez jedną cechę liczbową, gdzie 0 oznacza jedną z klas a 1 drugą.
- Cechy zawierające więcej kategorii (finishing_condition i address_estate) zostały zakodowane wykorzystując one hot encoding aby nie traktować kategorii w sposób ciągły.

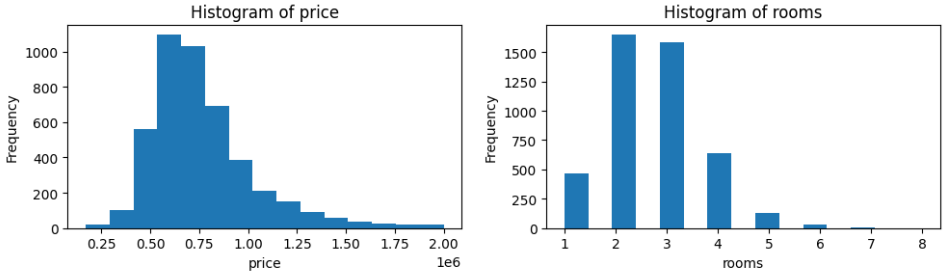
Wyizolowana została cecha price jako cel predykcji. Ze względu na stabilność modeli postanowiliśmy przewidywać cenę za metr kwadratowy, zamiast całkowitej ceny mieszkania.

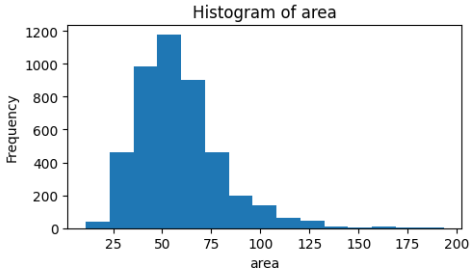
Zbiór został podzielony na 3 podzbiory - treningowy, walidacyjny i testowy, w celu uniknięcia zakłamania wyniku końcowego przy dopasowaniu parametrów do najkorzystniejszych wyników, konkretnego zbioru.

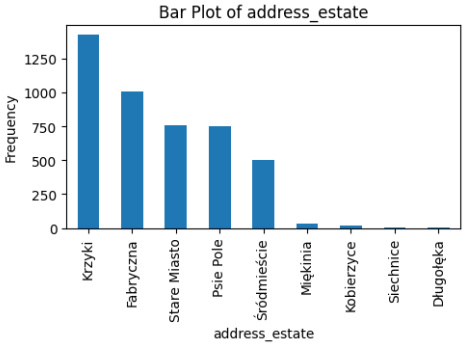
Tak przygotowany zbiór danych został przeskalowany z wykorzystaniem standardowego skalera do średnia=0 oraz odchylenie standardowe=1, dopasowanego do zbioru treningowego (zarówno dla cech wejściowych jak i dla celu).

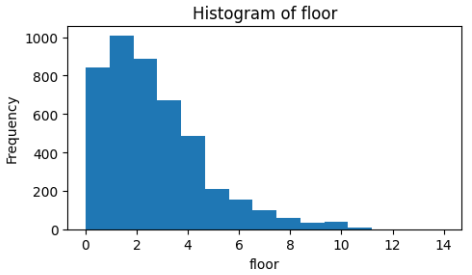
4 Eksploracja danych

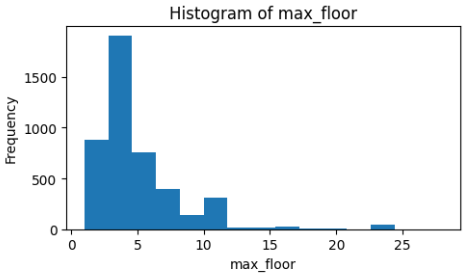
Histogramy wyczyszczonych danych przestawiają się w sposób następujacy:

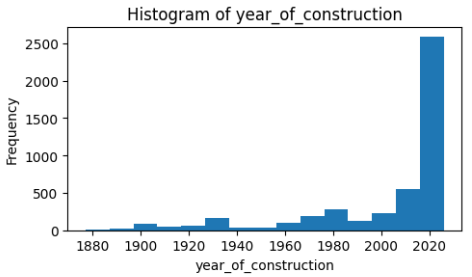


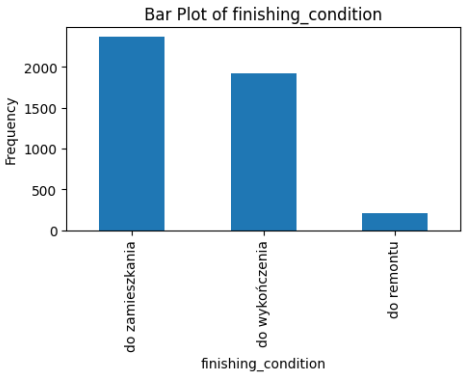




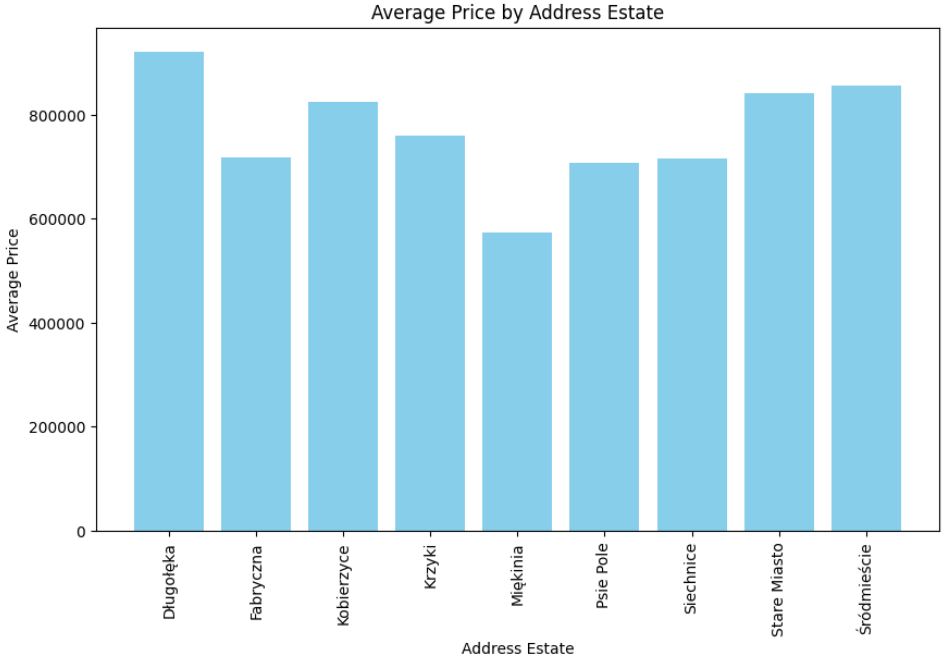




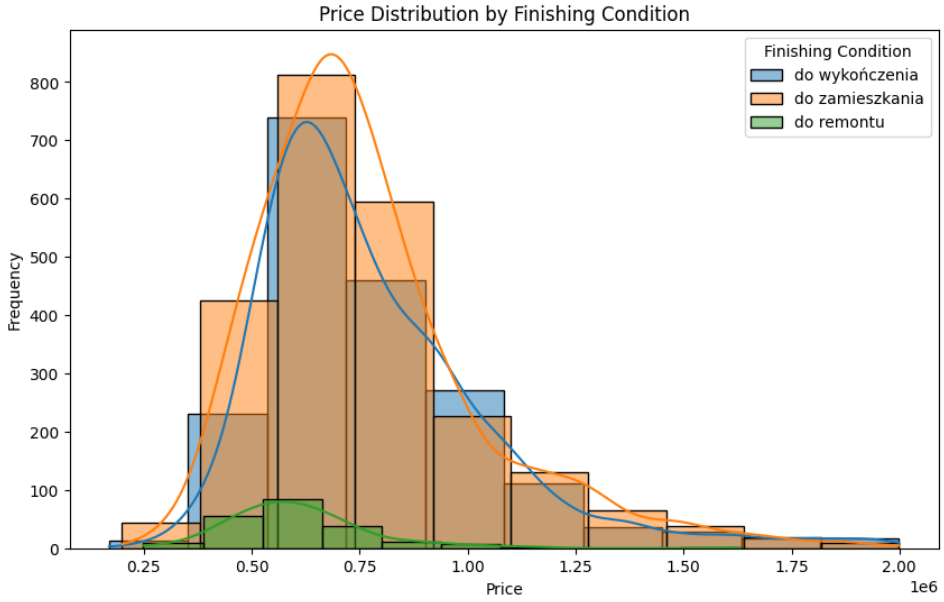


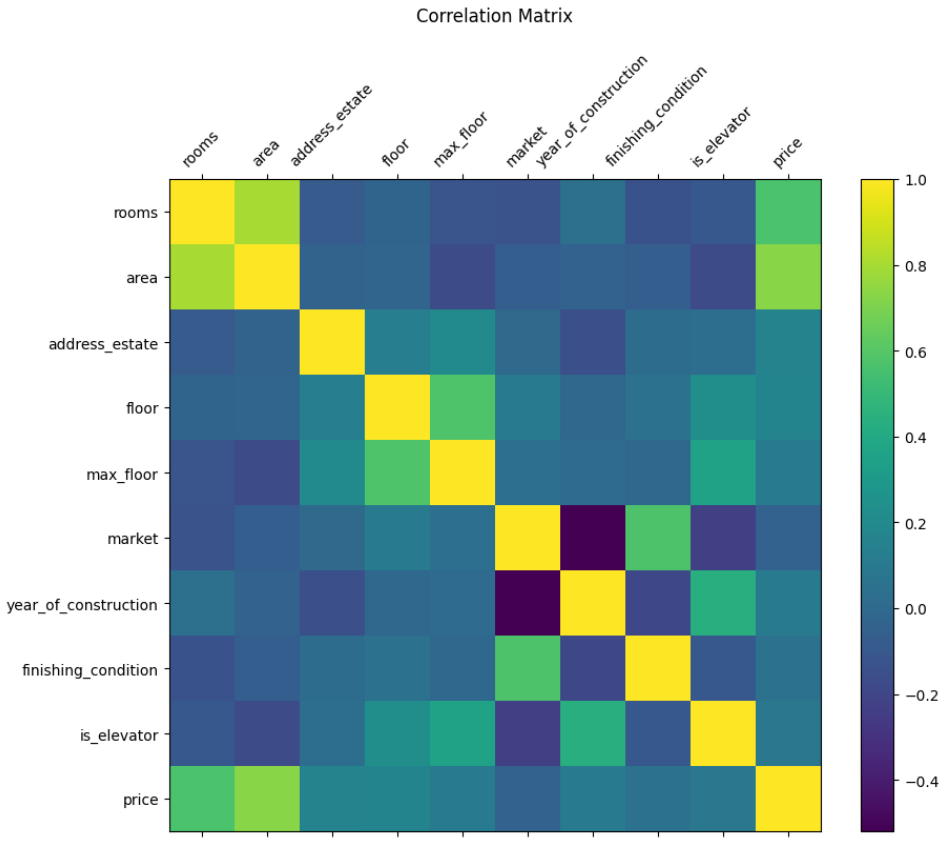


Przyjrzyjmy się średniej cenie nieruchomości według osiedla:



Zwróćmy też uwagę na cenę według stanu wykończenia. Wyraźnie mieszkania w stanie "do remontu" są najtańsze, a mieszkania gotowe "do zamieszkania" są najdroższe.





Kolejnym współczynnikiem badania danych jest macierz korelacji.

- Możemy zaobserwować silną zależność proporcjonalności prostej dla powierzchni mieszkania i ilości pokoi wskazującą na to, że mieszkania o większej powierzchni mają zazwyczaj więcej pokoi.
- Kolejną korelacją jest zależność piętra od wysokości budynku (maksymalnego piętra). Ta miara, może być delikatnie zakłamana ze względu na brak danych o piętrach 10+, co zostało uśrednione przy wstępnym przetwarzaniu danych, lecz nie wątpliwie dla wyższych budynków wysokości piętra mieszkań będą wyższe.
- Występuje również zależność warunków wykończenia od roku produkcji. Mieszkania nowe często wymagają dodatkowego wykończenia.
- Macierz wskazuje na silną proporcjonalności odwrotną między rokiem budowy i rynkiem (wtórnym, pierwotnym). Wynika to z tego, że mieszkania nowe to rynek pierwotny a starsze mieszkanie na rynku pierwotnym praktycznie nie występują.
- Ostatnim czynnikiem wartym szczególnej uwagi jest zależność ceny od powierzchni. Ze względu na wiele czynników takich jak mniejsza rozbieżność wartości, jak i charakterystyka rynku nieruchomości, zadecydowaliśmy trenować model dla wyceny metra kwadratowego.

5 Wybór i trening modeli

Do predykcji wartości wybrałem omawiane na lekcji modele regresji liniowej, uogólniony model liniowy (regresja wielomianem) oraz postanowiłem we własnym zakresie zgłębić wiedzę na temat sieci neuronowych i zastosować Multi Layer Perceptron.

Do implementacji modelu regresji liniowej i uogólnionego modelu liniowego został użyty moduł sklearn, a do implementacji sieci neuronowej moduł PyTorch.

5.1 Regresja liniowa

Celem modelu jest możliwie dokładnie dopasowanie prostej do punktów tak aby minimalizować kwadraty odchyleń przewidywanych wartości od rzeczywistych. W interpretacji geometrycznej prosta ta jest wielowymiarowa w zależności od ilości parametrów. Model działa bardzo dobrze w przypadku zależności liniowych między danymi, lecz nie jest w stanie dokładnie odwzorować bardziej złożonych zależności.

5.2 Uogólniony model liniowy

Wybór padł na ogólniony model liniowy z użyciem wielomianu. Rozszerza on standardową regresję liniową, dopasowując wielomian zamiast prostą. Pozwala to lepiej przybliżyć nieliniowe zależności między zmiennymi. W przypadku użycia zbyt wysokiego współczynnika wielomianu (jako parametru) model ma tendencję do zbytniego dopasowania do danych treningowych.

5.3 Sieć neuronowa

Sieci neuronowe to szerokie pojęcie. Uwaga została skupiona na modelu wielowarstwowym MLP (Multi-Layer Percepron). MLP składa się z trzech głównych rodzajów warstw: warstwy wejściowej, warstw ukrytych oraz warstwy wyjściowej. Każdy neuron w jednej warstwie jest połączony z neuronami w następnej warstwie za pomocą ważonych połączeń. Neurony przetwarzają dane wejściowe przez sumowanie ważonych wejść i przekazywanie wyniku przez funkcję aktywacji, która decyduje o aktywacji neuronu. Ta struktura pozwala MLP na uczenie się złożonych wzorców w danych poprzez dostosowywanie wag w procesie, znanym jako uczenie z nadzorem.

Architektura wybranej sieci składała się z następujących warstw:

1. Warstwa wejściowa - 19 neuronów
2. Warstwa ukryta 1 - 256 neuronów z funkcją aktywacji ReLU
3. Warstwa ukryta 2 - 128 neuronów z funkcją aktywacji ReLU
4. Warstwa wyjściowa - 1 neuron

Zastosowano gęstrze połączenia między kolejnymi warstwami. Została użyta funkcja aktywacji ReLU ze względu na jej niski koszt obliczeniowy. Po każdej z warstw ukrytych, na czas treningu, zastosowano dropout wielkości 30% każdy w celu uniknięcia nadmiernego dopasowania do zbioru treningowego.

Do treningu został użyty algorytm optymalizacji "Adam" ze współczynnikiem uczenia 0.001. Główna pętla ucząca została wykonana 70 razy (dla tego samego dataset'u). Za każdym powtórzeniem odbywała się optymalizacja współczynników sieci na podstawie całego zbioru danych, pogrupowanego w serie po 16 próbek, w celu przyspieszenia obliczeń i lepszemu uogólnianiu. Wykorzystana funkcja straty to błąd średniokwadratowy.

6 Porównanie modeli

Modele były testowane dla różnych parametrów i walidowane ze zbiorem testowym jak i walidacyjnym w celu uniknięcia zbytniego dopasowania do danych. Wyniki prezentują się następująco:

6.1 Regresja liniowa

Miara	**Zbiór Treningowy**	**Zbiór Walidacyjny**
Średni błąd kwadratowy	0.4893	0.5224
Średni błąd dla m^2	1693.7538	1734.4692
Odchylenie standardowe dla m^2	1648.1920	1718.8326
Średni błąd dla ceny końcowej	97528.945	103545.805
Odchylenie standardowe dla ceny końcowej)	104316.18	119547.67
Średni błąd procentowy	12.42%	12.67%
Odchylenie standardowe procentowe	11.51%	10.66%

6.2 Uogólniony model liniowy

Głównym parametrem możliwym do dostosowania w uogólnionym modelu liniowym za pomoca wielomianu jest jego stopień. Dla wszystkich przypadków biasa jest włączony.

Poniżej zestawienie modelu z różnymi parametrami zaczynając od stopnia wielomianu równego 2.

Miara	**Zbiór Treningowy**	**Zbiór Walidacyjny**
Średni błąd kwadratowy	0.3525	0.6407
Średni błąd dla m^2	1439.0934	1557.1069
Odchylenie standardowe dla m^2	1397.3699	2211.2212
Średni błąd dla ceny końcowej	82804.42	95302.6
Odchylenie standardowe dla ceny końcowej	90989.34	158313.98
Średni błąd procentowy	10.57%	11.77%
Odchylenie standardowe procentowe	10.80%	19.96%

Poniżej wyniki dla wielomianu 3. stopnia. Można zaobserwować wyraźnie lepszy wynik na zbiorze testowym co wskazuje na dopasowanie modelu do zbioru. Mimo tego wyniki na zbiorze walidacyjnym również się poprawiły, więc oceniamy ten wynik na plus.

Miara	**Zbiór Treningowy**	**Zbiór Walidacyjny**
Średni błąd kwadratowy	0.2618	0.5060
Średni błąd dla m^2	1210.2157	1522.0884
Odchylenie standardowe dla m^2	1234.6095	1859.8328
Średni błąd dla ceny końcowej	68576.68	92130.23
Odchylenie standardowe dla ceny końcowej	75832.54	145173.08
Średni błąd procentowy	8.83%	11.38%
Odchylenie standardowe procentowe	9.15%	14.68%

Gorzej ma się sprawa przy dopasowaniu do wielomiany 4. stopnia. Tutaj model jest wyraźnie przetrenowany i nie przynosi to żadnych korzyści, dane możemy zaobserwować poniżej:

Miara	Zbiór Treningowy	Zbiór Walidacyjny
Średni błąd kwadratowy	0.1981	20.5863
Średni błąd dla m^2	1068.9783	4009.1367
Odchylenie standardowe dla m^2	1057.8229	14796.1140
Średni błąd dla ceny końcowej	60033.36	245685.00
Odchylenie standardowe dla ceny końcowej	63752.855	921158.06
Średni błąd procentowy	7.89%	31.15%
Odchylenie standardowe procentowe	7.70%	111.58%

Za najatrakcyjniejszy z tych modeli uznaję model z parametrem 3. stopnia wielomianu.

6.3 Sieć neuronowa

Model sieci neuronowej pozwala na dostosowanie dużej ilości parametrów takich jak: jej warstwy, ilość, gęstość, funkcję aktywacji, specjalne działanie taki jak dropout'y jak i marametry uczenia takie jak: wielkość batch'a, współczynnik uczenia czy ilość epok.

Zostały przeprowadzone testy dla różnych kombinacji, kombinacja o najlepszych znalezionych parametrach uczenia posiada następujące wartości:

- batch size = 16
- learning rate = 0.001
- epochs = 70

Miara	Zbiór Treningowy	Zbiór Walidacyjny
Średni błąd kwadratowy	0.1999	0.3204
Średni błąd dla m^2	1074.6442	1296.0255
Odchylenie standardowe dla m^2	1061.7152	1406.3668
Średni błąd dla ceny końcowej	62666.812	76946.21
Odchylenie standardowe dla ceny końcowej	70112.96	96269.19
Średni błąd procentowy	8.12%	9.55%
Odchylenie standardowe procentowe	8.49%	9.18%

6.4 Ostateczne porównianie wszystkich modeli dla testowego zbioru danych

Miara	LIN	GLM	MLP
Średni błąd kwadratowy	0.5513	0.4685	0.3253
Średni błąd dla m^2	1762.2479	1538.9607	1277.5070
Odchylenie standardowe dla m^2	1785.525	1598.2399	1432.7299
Średni błąd dla ceny końcowej	99015.6797	89461.9375	70559.1094
Odchylenie standardowe dla ceny końcowej	100033.3750	118136.5000	77169.5859
Średni błąd procentowy	12.44%	10.97%	8.79%
Odchylenie standardowe procentowe	10.25%	10.91%	7.92%

Tabela 2: Wyniki badań skuteczności modeli

7 Wnioski

7.1 Regresja liniowa

Pomimo swojej prostoty, regresja liniowa okazała całkiem skutecznym modelem. W wynikach końcowych możemy dostrzec, że ma ona mniejsze odchylenie standardowe od modelu uogólnionego dla wielomianiu 3. stopnia, a średni błąd nie odbiega znacząco od pozostałych modeli. Jest to model odpowiedni do szybkiej, wstępnej analizy, ale nie odwzorowywuje dokładnie złożonych zależności.

7.2 Uogólniony model liniowy (uogólniony wielomianem)

Model ten ma tendencję do łatwego przetrenowywania, uzyskał średni błąd mniejszy od regresji liniowej.

7.3 Sieć neuronowa

Sieć neuronowa okazała się najskuteczniejsza spośród wszystkich testowanych modeli. Osiągnęła najniższe wartości błędów i największą stabilność, co czyni ją najlepszym wyborem do predykcji w tym przypadku. Ważnym czynnikiem, zapobiegającym przetrenowaniu było dropouty usuwające część wartości.

8 Podsumowanie

Rynek mieszkań charakteryzuje się dynamicznością i różnorodnością, codziennie pojawiają się w internecie setki nowych ofert. Wykorzystanie modeli uczenia maszynowego do szacowania cen nieruchomości, jak wskazuje powyższa praca, umożliwia ocenę atrakcyjności poszczególnych ogłoszeń. Takie podejście pozwala na lepsze dopasowanie ofert do oczekiwań klientów, co stanowi ogromną wartość zarówno dla kupujących, jak i sprzedających. Dzięki temu można szybciej i trafniej dokonywać wyborów, minimalizując ryzyko zakupu nieruchomości po zawyżonej cenie. W efekcie, nowoczesne narzędzia analityczne przyczyniają się do większej transparentności i efektywności na rynku nieruchomości.