Praca Dyplomowa Inżynierska

Jerzy Marczewski
214414

Prognoza cen nieruchomości przy pomocy uczenia maszynowego

Predicting Housing Prices with Machine Learning

Praca dyplomowa na kierunku:
Informatyka

Praca wykonana pod kierunkiem
dra hab. Konrada Furmańczyka, prof. SGGW
Instytut Informatyki Technicznej
Katedra Zastosowań Matematyki

Warszawa, rok 2024

SZKOŁA GŁÓWNA GOSPODARSTWA WIEJSKIEGO

Wydział Zastosowań Informatyki i Matematyki

Oświadczenie Promotora pracy

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data ..... Podpis promotora .....

Oświadczenie autora pracy

Świadom/a odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności z ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2019 poz. 1231 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną. Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiatowej.

Data ..... Podpis autora pracy .....

Streszczenie

Prognoza cen nieruchomości przy pomocy uczenia maszynowego

Tematem niniejszej pracy było zbudowanie modelu uczenia maszynowego, który jak najlepiej jest w stanie prognozować ceny nieruchomości na podstawie ich charakterystyk. W pracy przeprowadzono porównanie modeli opartych na różnych metodach uczenia maszynowego, takich jak: regresja liniowa, regresja grzbietowa, Lasso, elastic net, k najbliższych sąsiadów, maszyny wektorów nośnych, drzewa decyzyjne, lasy losowe, AdaBoost oraz gradient boosting. Praca przedstawia proces budowy modeli uczenia maszynowego oraz różne techniki budowy takich modeli. Porównane zostały modele trenowane na filtrowanych danych z modelami trenowanymi na niefiltrowanych danych. Podczas ostatecznej ewaluacji na danych testowych pokazano, że modele trenowane na danych niefiltrowanych uzyskiwały znacznie lepsze wyniki niż modele trenowane na danych filtrowanych. Najlepsze wyniki uzyskały modele gradient boosting oraz lasy losowe.

Słowa kluczowe – nieruchomości, uczenie maszynowe, analiza danych, inżynieria cech, filtrowanie, metody uczenia maszynowego, modele regresji, optymalizacja modeli, ewaluacja modeli

Summary

Predicting Housing Prices with Machine Learning

The subject of this thesis was to build a machine learning model capable of predicting housing prices based on their characteristics. The study compared models created using various machine learning methods such as linear regression, ridge regression, Lasso, elastic net, k-nearest neighbors, support vector machines, decision trees, random forests, AdaBoost, and gradient boosting. The thesis presents the process of building machine learning models and various techniques for constructing such models. Models trained on filtered data were compared with models trained on unfiltered data. The final evaluation on test data demonstrated that models trained on unfiltered data achieved significantly better results than those trained on filtered data. The best results were achieved by the gradient boosting and random forests models.

Keywords – real estate, machine learning, data analysis, feature engineering, filtering, machine learning methods, regression models, model optimization, model evaluation

Spis treści

1 Wstęp

1.1 Wprowadzenie do tematu

Nieruchomości pełnią funkcję zarówno miejsca zamieszkania, którego posiadanie jest jedną z podstawowych potrzeb człowieka, jak i formy inwestycji, która może generować dochód w postaci pobieranego czynszu z najmu lub w postaci zysku generowanego poprzez sprzedaż nieruchomości po wzroście jej wartości w czasie. Precyzyjna ocena wartości nieruchomości jest kluczowym elementem podejmowania świadomych decyzji na rynku nieruchomości, którego uczestnikami są inwestorzy i osoby pragnące zakupić miejsce zamieszkania. Wycena nieruchomości stanowi wyzwanie ze względu na mnogość czynników wpływających na ich wartość, obejmujących lokalizację, metraż, stan techniczny oraz inne istotne aspekty. Brak doświadczenia na rynku nieruchomości może znacznie utrudnić dokładną ocenę wartości nieruchomości. Rozwiązaniem tego problemu może być zastosowanie uczenia maszynowego. Istnieje wiele technik uczenia maszynowego, których modele mogą być trenowane na danych zawierających cechy nieruchomości wraz z odpowiadającymi im cenami. Gdy te modele zostaną wytrenowane, będą mogły prognozować wartości nieruchomości na podstawie nowo podanych wartości cech. Posiadając taki model, osoby będą mogły podejmować świadome decyzje na rynku nieruchomości, nawet nie posiadając wieloletniego doświadczenia, ponieważ po dostarczeniu charakterystyk do modelu otrzymają prognozowaną wartość nieruchomości.

1.2 Przegląd literatury

Tematyka prognozowania cen nieruchomości przy użyciu technik uczenia maszynowego cieszy się dużym zainteresowaniem w literaturze naukowej. Niniejsza praca wyróżnia się przede wszystkim tym, że przeprowadza porównanie dziesięciu różnych metod uczenia maszynowego w kontekście prognozowania cen nieruchomości. Dodatkowo, wykorzystuje dwa zróżnicowane zbiory danych, co pozwala na analizę wpływu charakterystyk danych na efektywność modeli oraz przeprowadza proces filtracji danych treningowych, co pozwala na porównanie skuteczności modeli w zależności od

zbioru treningowego. Poniżej przedstawione zostały prace naukowe związane z tematyką prognozowania cen nieruchomości przy użyciu technik uczenia maszynowego.

Praca [2] wykorzystuje zbiór danych „AmesHousing", który dostarcza informacji dotyczących nieruchomości zlokalizowanych w mieście Ames, w stanie Iowa, USA, wraz z odpowiadającymi im cenami. W zakresie analizy zostały zastosowane metody zarówno regresji, jak i klasyfikacji. W kontekście regresji zostały zaimplementowane różnorodne techniki, takie jak Lasso, regresja grzbietowa, SVM oraz lasy losowe. Natomiast w przypadku klasyfikacji, ceny nieruchomości zostały skategoryzowane na przedziały cenowe, a prognozy dokonano przy użyciu metod klasyfikacyjnych, w tym naive bayes, regresja logistyczna, SVM i lasy losowe. Głównym celem przedstawionej pracy było stworzenie modeli regresji i klasyfikacji, które umożliwią precyzyjne oszacowanie wartości nieruchomości, uwzględniając zróżnicowane cechy charakteryzujące analizowane nieruchomości. Zarówno dla problemu klasyfikacji, jak i regresji, wyniki wskazują, że model SVM uzyskał najwyższą skuteczność w prognozowaniu cen nieruchomości.

Praca [3] wykorzystuje zbiór danych „Housing Price in Beijing", zawierający informacje dotyczące nieruchomości zlokalizowanych w Pekinie. Badanie korzysta zarówno z tradycyjnych, jak i zaawansowanych metod uczenia maszynowego, analizując różnice między zaawansowanymi modelami. W ramach eksperymentu użyto metod, takich jak lasy losowe, XGBoost i LightGBM, oraz techniki uczenia maszynowego, w tym Hybrid Regression oraz Stacked Generalization Regression. Zgodnie z wynikami pracy, metoda Stacked Generalization Regression, choć charakteryzuje się złożoną architekturą, została uznana za najlepszy wybór.

Praca [4] oparta jest na danych z 1970 roku dotyczących nieruchomości w Bostonie. W pracy zastosowano różnorodne metody uczenia maszynowego, takie jak regresja liniowa, lasy losowe, XGBoost i SVM. Badanie wykazało, że zaawansowane modele uczenia maszynowego przewyższają tradycyjne modele regresji liniowej. Spośród czterech modeli, XGBoost wykazał się najwyższą skutecznością w przewidywaniu cen nieruchomości. Praca dowodzi, że nowoczesne techniki uczenia maszynowego są bardziej efektywne niż klasyczne podejścia regresyjne, co ma istotne znaczenie w kontekście prognozowania cen mieszkań.

2 Cel i Zakres Pracy

Celem niniejszej pracy jest przeprowadzenie analizy oraz porównania skuteczności różnych modeli uczenia maszynowego w kontekście prognozowania cen nieruchomości. Badania oparte są na zestawach danych z dwóch różnych regionów, charakteryzujących się zróżnicowanymi cechami. Zakres pracy obejmuje analizę danych oraz ich przetwarzanie, w tym identyfikację oraz obsługę obserwacji odstających, inżynierię cech oraz filtrację danych, mającą na celu przygotowanie zbiorów treningowych do kolejnych etapów badania. W ramach procesu modelowania zastosowano kroswalidację w celu wyboru najbardziej obiecujących modeli, które zostały następnie poddane trenowaniu i ewaluacji. Głównym celem jest identyfikacja najbardziej efektywnych modeli oraz zrozumienie wpływu różnych operacji przetwarzania danych na ich skuteczność w prognozowaniu cen nieruchomości.

3 Opis Założeń Metodycznych

W niniejszym rozdziale omówione zostały kluczowe aspekty metodologiczne związane z przeprowadzeniem badania. Obejmują one wybór, analizę i przetwarzanie zestawów danych, opis wybranych metod uczenia maszynowego, a także opis procesu budowy modeli i ich ewaluacji.

3.1 Wybór danych

W pracy zostały wykorzystane dwa zbiory danych. Nazwy cech w zbiorach zostały zmienione, aby przestrzegały tej samej konwencji.

Pierwszym z wykorzystanych zbiorów jest zbiór „California Housing", dostępny w bibliotece scikit-learn [6], z oryginalnym źródłem w repozytorium StatLib [7]. Zbiór ten pochodzi ze spisu ludności USA z 1990 roku i zawiera 20 640 obserwacji. Każdy wiersz reprezentuje dane dotyczące pojedynczej grupy bloków (ang. block group), będącej najmniejszą geograficzną jednostką, dla której U.S. Census Bureau publikuje dane.

Tabela 3.1. Tabela przedstawia opisy cech zbioru „California Housing". Opracowanie własne.

Nazwa cechy	**Opis**
MED_INC	Mediana zarobków (w dziesiątkach tysięcy dolarów)
HOUSE_AGE	Mediana wieku budynków
AVE_ROOMS	Średnia liczba pokoi na gospodarstwo domowe
AVE_BEDRMS	Średnia liczba sypialni na gospodarstwo domowe
POPULATION	Liczba mieszkańców
AVE_OCCUP	Średnia liczba domowników na gospodarstwo domowe
LATITUDE	Szerokość geograficzna
LONGITUDE	Długość geograficzna
MED_HOUSE_VAL	Mediana wartości nieruchomości (w setkach tysięcy dolarów)

Drugim zbiorem danych wykorzystanym w badaniu jest zbiór „MiamiHousing2016", stworzony przez Profesora Stevena C. Bourassa. Zawiera on 13 932 obserwacje dotyczące transakcji sprzedaży domów jednorodzinnych w obszarze metropolitalnym Miami z 2016 roku. Zbiór jest dostępny w ramach repozytorium OpenML [8]. W dalszej części pracy zbiór będzie określany jako „Miami Housing".

Tabela 3.2. Tabela przedstawia opisy cech zbioru „Miami Housing". Opracowanie własne.

Nazwa cechy	**Opis**
PARCEL_NO	Identyfikator posiadłości
LND_SQFOOT	Powierzchnia działki (w stopach kwadratowych)
TOT_LVG_AREA	Powierzchnia użytkowa domu (w stopach kwadratowych)
SPEC_FEAT_VAL	Wartość dodatkowych udogodnień (w dolarach)
RAIL_DIST	Odległość do najbliższej linii kolejowej (w stopach)
OCEAN_DIST	Odległość do oceanu (w stopach)
WATER_DIST	Odległość do najbliższego zbiornika wodnego (w stopach)
CNTR_DIST	Odległość do centralnej biznesowej dzielnicy w Miami (ang. Miami central business district) (w stopach)
SUBCNTR_DI	Odległość do najbliższego centrum handlowego (w stopach)
HWY_DIST	Odległość do najbliższej autostrady (w stopach).
AGE	Wiek budynku
AV_NO_60_PLUS	Zmienna sygnalizująca, że poziom hałasu generowany przez samoloty przekracza ustaloną akceptowalną wartość (0 lub 1)
STRUCTURE_QUALITY	Jakość konstrukcji (w skali od 1 do 5)
MONTH_SOLD	Miesiąc w którym dom został sprzedany (o zakresie od 1 do 12)
LATITUDE	Szerokość geograficzna
LONGITUDE	Długość geograficzna
SALE_PRC	Cena sprzedaży domu (w dolarach)

Warto podkreślić, że cecha PARCEL_NO nie będzie uwzględniana w analizach przeprowadzanych w ramach niniejszej pracy. Jest to cecha identyfikacyjna, której obecność jest zbędna w niniejszym badaniu.

Wykorzystane zbiory danych różnią się zarówno cechami, jak i strukturą. Zbiór „California Housing" przedstawia dane dotyczące grup bloków, podczas gdy zbiór „Miami Housing" skupia się na pojedynczych domach. Różnice w strukturze zbiorów danych mogą mieć istotny wpływ na skuteczność modeli uczenia maszynowego oraz na wyniki przeprowadzanych badań. Porównanie tych różnic pozwala ocenić, jak modele uczenia maszynowego zachowują się w zależności od charakterystyki danych, na których są trenowane.

3.2 Przygotowanie danych

W wykorzystanych zbiorach danych nie ma braków danych, a wszystkie cechy są reprezentowane w postaci numerycznej. Ten fakt jest istotnym atutem podczas procesu przygotowania danych, gdyż eliminuje potrzebę stosowania procedur obsługi brakujących danych oraz konwersji cech na format numeryczny. To przyspiesza etap

przygotowania danych i jednocześnie eliminuje potencjalne zakłócenia wynikające z konieczności tych operacji.

3.3 Analiza danych

W tej sekcji przedstawiony zostanie proces analizy danych. Wszelkie wymienione operacje zostają zastosowane zarówno dla zbioru „California Housing" jak i zbioru „Miami Housing". Aby lepiej zrozumieć strukturę i rozkład danych dotyczących każdej cechy w zbiorach, przedstawiono podstawowe charakterystyki opisowe, obejmujące liczbę rekordów, średnią, odchylenie standardowe, wartość minimalną, pierwszy kwartyl, drugi kwartyl, trzeci kwartyl oraz wartość maksymalną. Dodatkowo, rozkład tych danych został zobrazowany za pomocą histogramów.

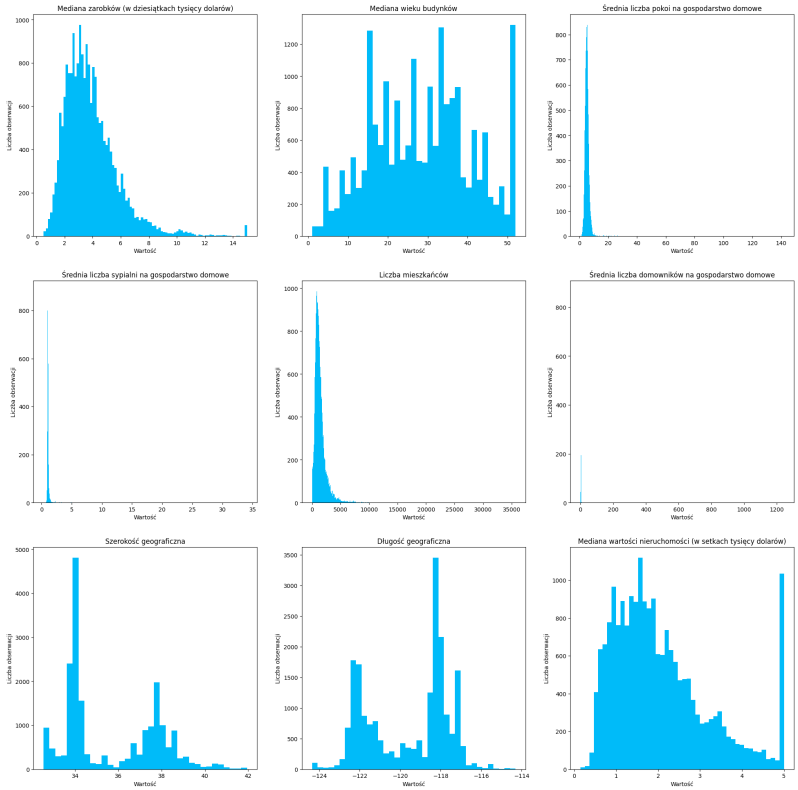
	MED_INC	HOUSE_AGE	AVE_ROOMS	AVE_BEDRMS	POPULATION	AVE_OCCUP	LATITUDE	LONGITUDE	MED_HOUSE_VAL
count	20640.0000	20640.0000	20640.0000	20640.0000	20640.0000	20640.0000	20640.0000	20640.0000	20640.0000
mean	3.8707	28.6395	5.4290	1.0967	1425.4767	3.0707	35.6319	-119.5697	2.0686
std	1.8998	12.5856	2.4742	0.4739	1132.4621	10.3860	2.1360	2.0035	1.1540
min	0.4999	1.0000	0.8462	0.3333	3.0000	0.6923	32.5400	-124.3500	0.1500
25%	2.5634	18.0000	4.4407	1.0061	787.0000	2.4297	33.9300	-121.8000	1.1960
50%	3.5348	29.0000	5.2291	1.0488	1166.0000	2.8181	34.2600	-118.4900	1.7970
75%	4.7432	37.0000	6.0524	1.0995	1725.0000	3.2823	37.7100	-118.0100	2.6472
max	15.0001	52.0000	141.9091	34.0667	35682.0000	1243.3333	41.9500	-114.3100	5.0000

Rysunek 3.1. Rysunek przedstawia tabele zawierającą podstawowe charakterystyki opisowe cech zbioru „California Housing". Opracowanie własne. Na tym oraz wszystkich kolejnych rysunkach liczby zmiennoprzecinkowe będą separowane kropką, zgodnie z konwencją stosowaną w języku Python. Przedstawione wartości zostały zaokrąglone do czterech miejsc po przecinku.

	LATITUDE	LONGITUDE	SALE_PRC	LND_SQFOOT	TOT_LVG_AREA	SPEC_FEAT_VAL	RAIL_DIST	OCEAN_DIST	WATER_DIST	CNTR_DIST	SUBCNTR_DI	HWY_DIST	AGE	AV_NO_60_PLUS	MONTH_SOLD	STRUCTURE_QUALITY
count	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000	13932.0000
mean	25.7288	-80.3275	399941.9317	8620.8799	2058.0446	9562.4935	8348.5487	31690.9938	11960.2852	68490.3271	41115.0473	7723.7707	30.6693	0.0149	6.6558	3.5140
std	0.1406	0.0892	317214.6838	6070.0887	813.5385	13890.9678	6178.0273	17595.0795	11932.9924	32008.4748	22161.8259	6068.9361	21.1531	0.1213	3.3015	1.0974
min	25.4343	-80.5422	72000.0000	1248.0000	854.0000	0.0000	10.5000	236.1000	0.0000	3825.6000	1462.8000	90.2000	0.0000	0.0000	1.0000	1.0000
25%	25.6201	-80.4033	235000.0000	5400.0000	1470.0000	810.0000	3299.4500	18079.3500	2675.8500	42823.1000	23996.2500	2998.1250	14.0000	0.0000	4.0000	2.0000
50%	25.7318	-80.3389	310000.0000	7500.0000	1877.5000	2765.5000	7106.3000	28541.7500	6922.6000	65852.4000	41109.9000	6159.7500	26.0000	0.0000	7.0000	4.0000
75%	25.8523	-80.2580	428000.0000	9126.2500	2471.0000	12352.2500	12102.6000	44310.6500	19200.0000	89358.3250	53949.3750	10854.2000	46.0000	0.0000	9.0000	4.0000
max	25.9744	-80.1197	2650000.0000	57064.0000	6287.0000	175020.0000	29621.5000	75744.9000	50399.8000	159976.5000	110553.8000	48167.3000	96.0000	1.0000	12.0000	5.0000

Rysunek 3.2. Rysunek przedstawia tabele zawierającą podstawowe charakterystyki opisowe cech zbioru „Miami Housing". Opracowanie własne. Przedstawione wartości zostały zaokrąglone do czterech miejsc po przecinku.

Analizując histogramy przedstawione na rysunkach 3.3 i 3.4, zauważalne jest, że rozkłady poszczególnych cech nie przypominają rozkładów normalnych. Ta obserwacja ma znaczenie z punktu widzenia zastosowania pewnych technik statystycznych, które zakładają normalność danych. Histogram zmiennej objaśnianej MED_HOUSE_VAL (znajdujący się w prawym dolnym rogu rysunku 3.3) ukazuje

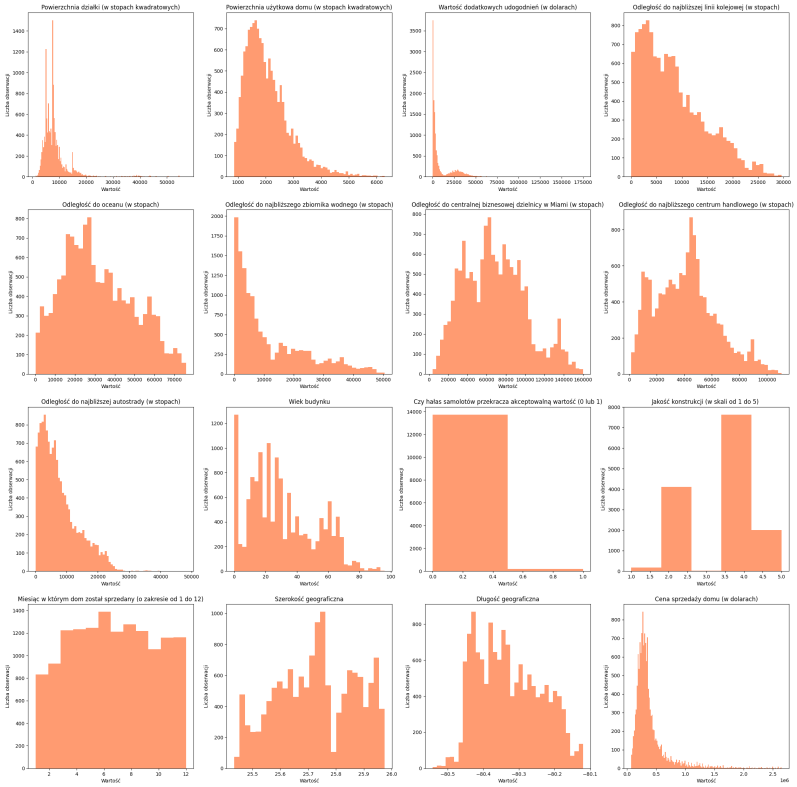


Rysunek 3.3. Rysunek przedstawia histogramy poszczególnych cech zbioru „California Housing". Opracowanie własne.

znaczną liczbę obserwacji skupionych wokół wartości 5. Istnieje prawdopodobieństwo, że ten nagły wzrost na wykresie wynika z formatowania danych w zbiorze, gdzie wartości tej zmiennej powyżej 5 mogą być ograniczone do tej wartości.

W celu oceny wzajemnych zależności między cechami skorzystano z Tau Kendalla. Współczynnik korelacji zwany Tau Kendalla został odkryty przez M.G. Kendalla w 1938 roku. Współczynnik jest nieparametryczny, a więc zakłada on minimalne założenia dotyczące rozkładu danych. Tau Kendalla jest miarą monotonicznej zależności między zmiennymi [22]. Jest to metoda rangowa, gdzie rozkład rang jest niezależny od zmiennych, które są poddane procesowi nadania rang, a więc ta metoda nie zakłada żadnych szczególnych założeń co do tego, jak dane są rozłożone, co sprawia, że jest odporna na nieregularności w rozkładzie danych. Tau Kendalla przyjmuje wartości w przedziale od -1 do 1 włącznie. Wartości współczynnika Tau Kendalla dla poszczególnych cech zbiorów zostały przedstawione w postaci macierzy.

Analizując rysunek 3.5 można zauważyć, że zmienna, która ma największą monotoniczną zależność ze zmienną objaśnianą MED_HOUSE_VAL jest zmienna MED_INC. Tau Kendalla dla tych zmiennych wynosi 0,5, co sugeruje umiarkowane powiązanie między zmiennymi, a dodatnia wartość współczynnika sugeruje dodatnią korelacje



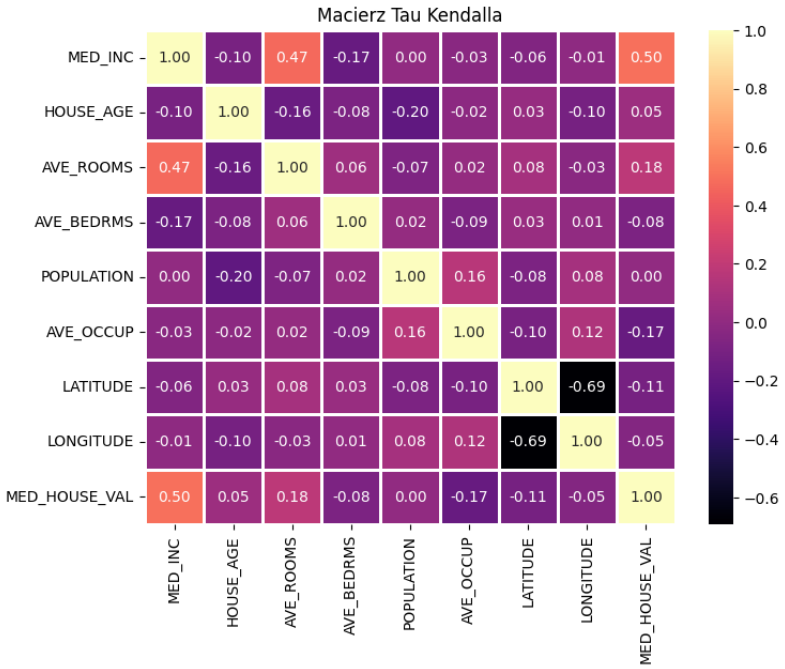
Rysunek 3.4. Rysunek przedstawia histogramy poszczególnych cech zbioru „Miami Housing". Opracowanie własne.

między zmiennymi. Na rysunku 3.6 jest zauważalne, że zmienna, która ma największą monotoniczą zależność ze zmienną objaśnianą SALE_PRC jest zmienna TOT_LVG_AREA. Tau Kendalla dla tych zmiennych wynosi 0,51, a więc można stwierdzić, że powiązanie miedzy zmiennymi jest umiarkowane, a korelacja między zmiennymi jest dodatnia.

W celu wizualizacji zmian wartości w zależności od współrzędnych geograficznych nieruchomości zastosowano wykres punktowy. Na wykresach każdy punkt reprezentuje jedną obserwację, a kolor punktów staje się coraz jaśniejszy wraz ze wzrostem wartości nieruchomości .

Na rysunku 3.7 można zauważyć dwa skupiska obserwacji nieruchomości o wysokich wartościach. Są to obszary metropolitalne San Francisco i Los Angeles. Ponadto, widoczne jest, że nieruchomości położone wzdłuż linii wybrzeża Kalifornii są zazwyczaj bardziej wartościowe od tych, które znajdują się z dala od wybrzeża.

Na rysunku 3.8 wyraźnie widać skupisko nieruchomości o wysokich wartościach, zlokalizowanych blisko środka obrazu. Jest to przybrzeżna część Coral Gables, miasta w metropolitalnym obszarze Miami. W tym rejonie znajdują się luksusowe osiedla, co tłumaczy, dlaczego ceny nieruchomości są tu wyższe niż w sąsiednich okoli-



Rysunek 3.5. Rysunek przedstawia macierz wartości Tau Kendalla dla poszczególnych cech zbioru „California Housing". Opracowanie własne. Przedstawione wartości zostały zaokrąglone do dwóch miejsc po przecinku.

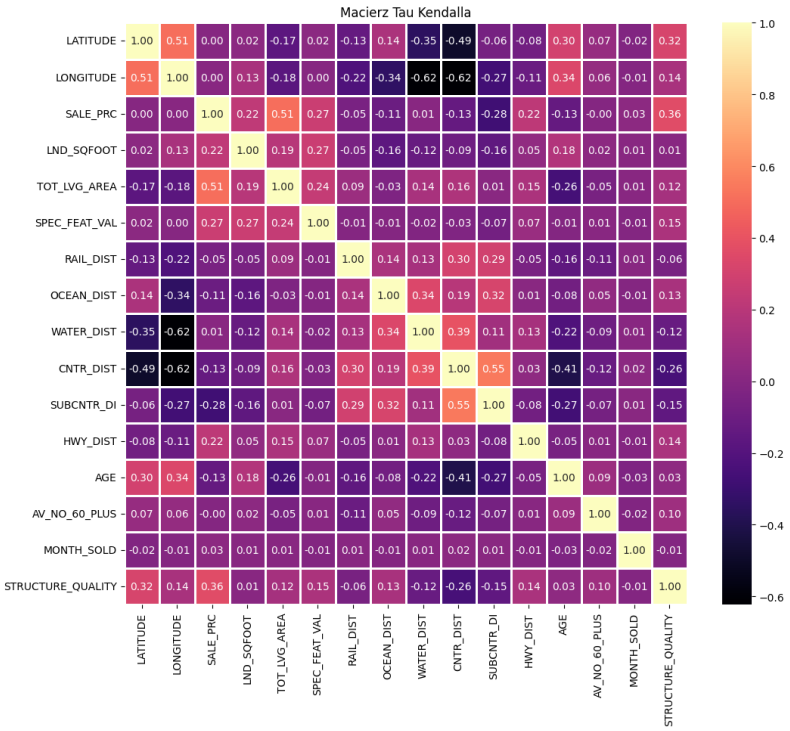
cach. W prawym górnym rogu rysunku widoczne są pojedyncze obserwacje reprezentujące nieruchomości o wysokich cenach, zlokalizowane na wyspach oddzielonych od kontynentu przez zatokę Biscayne. Ten obszar słynie z luksusowych plaż i hoteli.

3.4 Podział danych

Podział zbioru danych na zbiory treningowe i testowe pozwala na ocenę skuteczności modelu. Zbiór treningowy służy do uczenia modelu, natomiast zbiór testowy jest wykorzystywany do oceny jego wydajności. Prognozy generowane przez model na podstawie zbioru testowego są porównywane z rzeczywistymi wartościami z tego zbioru, co umożliwia ocenę tego, jak dobrze model radzi sobie z nowymi danymi [9]. W pracy zbiory danych zostały podzielone na zbiory treningowe i testowe w taki sposób, że zbiory testowe stanowią 25% wszystkich obserwacji w danym zbiorze.

3.5 Przetwarzanie danych

Dane ze zbiorów zostały poddane różnorodnym procesom przetwarzania w celu właściwego przygotowania ich do procesu modelowania.

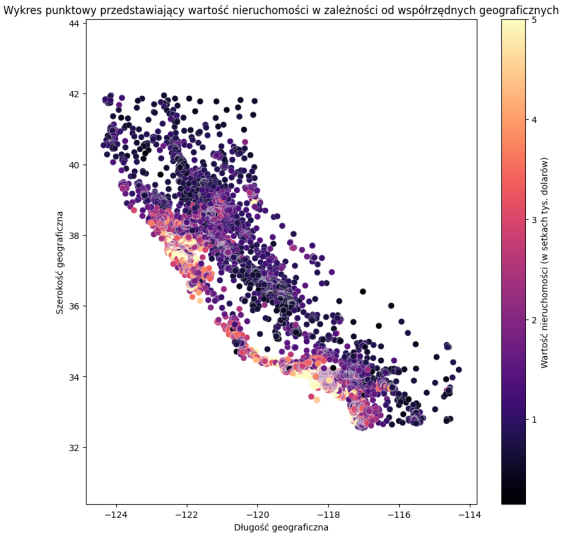


Rysunek 3.6. Rysunek przedstawia macierz wartości Tau Kendalla dla poszczególnych cech zbioru „Miami Housing". Opracowanie własne. Przedstawione wartości zostały zaokrąglone do dwóch miejsc po przecinku.

3.5.1 Obsługiwanie obserwacji odstających

Pierwszym z wykorzystanych procesów przetwarzania danych jest znajdywanie, a następnie usuwanie obserwacji odstających. Proces jest wykonywany jedynie na zbiorach treningowych, ponieważ obserwacje odstające mogą prowadzić do zakłóceń podczas procesu uczenia, co może objawiać się nieoptymalnie dopasowanym modelem. Dzięki temu, że obserwacje odstające nie są usuwane ze zbioru testowego, pozostaje on wiernym odzwierciedleniem rzeczywistych warunków, co umożliwia dokładniejszą ocenę wydajności modelu. W celu znalezienia obserwacji odstających w zbiorach treningowych „California Housing" i „Miami Housing" wykorzystano Isolation Forest.

Metoda Isolation Forest, często nazywana iForest, została zaproponowana przez Fei Tony Liu, Kai Ming Ting i Zhi-Hua Zhou w 2008 roku. Algorytm ten opiera się na ilościowych cechach anomalii, które stanowią mniejszość, oraz na ich właściwościach atrybutów, znacząco różniących się od tych występujących w przypadku normalnych instancji. Metoda polega na konstruowaniu struktury drzewa w celu izolacji każdej pojedynczej instancji. W tej strukturze drzewa anomalie są izolowane bliżej korzenia, ponieważ są one bardziej podatne na izolację, podczas gdy punkty normalne



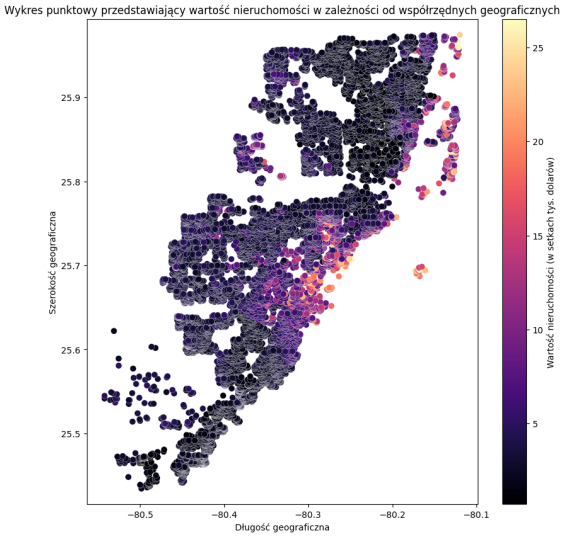
Rysunek 3.7. Rysunek przedstawia zmianę wartości nieruchomości ze zbioru „California Housing" wraz ze zmianą ich położenia. Opracowanie własne.

są izolowane głębiej w drzewie. Ta charakterystyka izolacji drzewa, nazwana przez autorów Isolation Tree lub iTree, stanowi podstawę metody wykrywania anomalii. Metoda Isolation Forest tworzy zbiór drzew izolacyjnych dla określonego zbioru danych i uznaje za anomalie te instancje, które mają krótkie średnie długości ścieżek na tych drzewach izolacyjnych [5]. Metoda Isolation Forest „izoluje" obserwacje poprzez losowe wybieranie cechy oraz losowe dobieranie wartości podziału pomiędzy maksymalną a minimalną wartością wybranej cechy. Ten proces rekurencyjnych podziałów może być przedstawiony za pomocą struktury drzewa, gdzie liczba podziałów potrzebnych do izolacji próbki odpowiada długości ścieżki od węzła korzenia do węzła końcowego [6].

3.5.2 Inżynieria cech

Drugim z wykorzystanych procesów przetwarzania danych jest inżynieria cech. Jest to technika w dziedzinie uczenia maszynowego, która polega na wykorzystywaniu danych do stworzenia nowych zmiennych, których nie ma w zestawie danych. Może generować nowe cechy zarówno dla uczenia nadzorowanego, jak i nienadzorowanego, mając na celu ułatwienie i przyspieszenie transformacji danych, jednocześnie poprawiając dokładność modelu [10].

Do danych zostały dodane nowe cechy, które są wynikiem ilorazów pierwotnych cech. Proces doboru tych ilorazów został przeprowadzony w taki sposób, aby uniknąć sytuacji, w której iloraz cech byłby wynikiem dzielenia cechy przez tę samą cechę lub



Rysunek 3.8. Rysunek przedstawia zmianę wartości nieruchomości ze zbioru „Miami Housing" wraz ze zmianą ich położenia. Opracowanie własne.

byłby odwrotnością innego ilorazu. W przypadku wystąpienia próby dzielenia przez zero podczas tworzenia cechy ilorazu ta konkretna cecha nie została uwzględniona.

3.5.3 Filtrowanie

Trzecim zastosowanym procesem przetwarzania danych jest filtrowanie. Polega ono na wyborze odpowiedniego wycinka zbioru danych. Operacja filtrowania wymaga ustalenia reguł identyfikujących konkretne przypadki, które spełniają określone kryteria porządku. Podobnie jak w przypadku obsługiwania obserwacji odstających, proces filtracji jest dokonywany jedynie na danych treningowych w celu zachowania rzeczywistego charakteru danych testowych.

W pracy celem filtrowania jest identyfikacja obserwacji reprezentujących nieruchomości charakteryzujące się umiarkowanymi rozmiarami. W przypadku zbioru danych „California Housing" zastosowano filtrację, aby uwzględnić jedynie te obserwacje, które charakteryzują się liczbą pokoi nieprzekraczającą 5. W przypadku zbioru danych „Miami Housing", z kolei uwzględniono obserwacje, których powierzchnia posesji nie przekracza 5500 stóp kwadratowych, a powierzchnia użytkowa nie przekracza 2500 stóp kwadratowych. Zbiory filtrowane w ten sposób mogą pomóc w lepszym dostosowaniu modelów do rzeczywistych warunków runku nieruchomości. W dalszej części pracy porównywane będą modele trenowane na danych filtrowanych z modelami trenowanymi na danych niefiltrowanych. Analiza ta pozwoli zrozumieć czy wykorzystana filtracja danych wpływa na skuteczność modeli.

3.5.4 Skalowanie cech

Skalowanie cech to technika używana do unormowania zakresu niezależnych cech. Istnieje wiele metod skalowania, lecz w pracy została wykorzystana standaryzacja.

Standaryzacja (zwana też „z-score normalization") polega na przeskalowaniu każdej cechy w taki sposób, aby średnia tej cechy wynosiła 0, a jej odchylenie standardowe wynosiło 1. Na niektóre modele jak na przykład na drzewa decyzyjne skalowanie praktycznie nie ma wpływu, lecz na wiele modeli ma diametralny wpływ. Przykładem takiego modelu jest model k najbliższych sąsiadów. Z racji, że model liczy odległość euklidesową między punktami, to w sytuacji, gdy jedna z cech ma znacznie większy zakres wartości, ma ona nieproporcjonalnie większy wpływ na obliczaną odległość [11]. Standaryzację można wyrazić następującym wzorem:

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

, gdzie:

- x - zmienna niestandaryzowana
- μ - średnia cechy
- σ - odchylenie standardowe cechy

Źródło: [12]

3.6 Wybrane metody uczenia maszynowego

W tej sekcji przedstawione zostaną metody uczenia maszynowego wykorzystane w pracy.

3.6.1 Regresja liniowa

Regresja liniowa wykorzystująca metodę najmniejszych kwadratów to najprostsza metoda rozwiązywania problemów regresji. Celem metody jest znalezienie hiperpłaszczyzny minimalizującej sumę kwadratów błędów (ang. SSE - sum-of-squared errors) między wartościami obserwowanymi, a wartościami prognozowanymi [20]. Po znalezieniu optymalnej hiperpłaszczyzny można odczytać jej parametry. Pierwszym parametrem jest wyraz wolny (ang. intercept). Jest to wartość, jaką przyjmuje zmienna zależna, kiedy wszystkie predyktory są równe zeru. Drugim parametrem

jest wektor współczynników (ang. coefficients). Wektor ten zawiera współczynniki określające wpływ poszczególnych predyktorów na zmienną zależną [6]. Postać liniową regresji można przedstawić następująco:

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon_i \tag{3.2}$$

, gdzie:

- y_i reprezentuje numeryczną odpowiedź dla i-tej próbki
- β_0 to wyraz wolny
- β_j reprezentuje szacowany współczynnik dla j-tego predyktora
- X_{ij} reprezentuje wartość j-tego predyktora dla i-tej próbki
- p reprezentuje liczbę predyktorów
- ϵ_i reprezentuje błąd

Źródło: [20]

Sumę kwadratów błędów można zapisać za pomocą poniższego wzoru:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3.3}$$

, gdzie:

- n oznacza liczbę obserwacji w zbiorze danych
- y_i to rzeczywista wartość dla i-tej obserwacji
- $\hat{y}_i$ to przewidywana wartość dla i-tej obserwacji przez model regresji liniowej

Źródło: [20]

3.6.2 Regresja grzbietowa

Regresja grzbietowa to metoda rozwiązująca problem zawyżonych współczynników wyznaczonych przez regresję liniową, które mogą być wynikiem nadmiernego dopasowania modelu regresji liniowej do danych. Regresja grzbietowa często jest wykorzystywana, gdy predyktory są silnie skorelowane. Metoda kontroluje wielkość współczynników poprzez nadanie kary L_2 podczas obliczania sumy kwadratów błędów. Efektem nałożenia tej kary jest zezwolenie na duże wartości współczynników jedynie, gdy zachodzi proporcjonalne zmniejszenie sumy kwadratów błędów [20]. Wyrażenie, które minimalizuje metoda, będące zmodyfikowaną sumą kwadratów

błędów można zapisać w sposób:

$$SSE_{L_2} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3.4}$$

, gdzie:

- n oznacza liczbę obserwacji w zbiorze danych
- y_i to rzeczywista wartość dla i-tej obserwacji
- $\hat{y}_i$ to przewidywana wartość dla i-tej obserwacji przez model regresji
- λ kontroluje siłę nałożonej kary
- p reprezentuje liczbę predyktorów
- β_j reprezentuje współczynnik regresji dla j-tego predyktora

Źródło: [20]

3.6.3 Lasso

Lasso (least absolute shrinkage and selection operator) to metoda, która tak jak regresja grzbietowa kontroluje wielkość współczynników poprzez nadanie kary. Metoda nadaje karę L_1 podczas obliczania sumy kwadratów błędów [20]. Wyrażenie, które minimalizuje metoda będące zmodyfikowaną sumą kwadratów błędów można zapisać następująco:

$$SSE_{L_1} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.5}$$

, gdzie:

- n oznacza liczbę obserwacji w zbiorze danych
- y_i to rzeczywista wartość dla i-tej obserwacji
- $\hat{y}_i$ to przewidywana wartość dla i-tej obserwacji przez model regresji
- λ kontroluje siłę nałożonej kary
- p reprezentuje liczbę predyktorów
- β_j reprezentuje współczynnik regresji dla j-tego predyktora

Źródło: [20]

3.6.4 Elastic net

Metoda elastic net łączy dwa typy kar wykorzystywanych w regresji grzbietowej i metodzie Lasso [20]. Wyrażenie, które minimalizuje metoda, będące zmodyfikowaną

sumą kwadratów błędów można wyrazić w następujący sposób:

$$SSE_{Enet} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} \beta_j^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j| \tag{3.6}$$

, gdzie:

- n oznacza liczbę obserwacji w zbiorze danych
- y_i to rzeczywista wartość dla i-tej obserwacji
- $\hat{y}_i$ to przewidywana wartość dla i-tej obserwacji przez model regresji
- λ_1, λ_2 kontrolują siły nałożonych kar
- p reprezentuje liczbę predyktorów
- β_j reprezentuje współczynnik regresji dla j-tego predyktora

Źródło: [20]

3.6.5 K najbliższych sąsiadów

Algorytm k najbliższych sąsiadów jest klasyfikatorem uczonym w sposób nadzorowany. Jest on nieparametryczny, a więc model nie przyjmuje założeń dotyczących zależności między zmiennymi objaśniającymi, a zmienna objaśnianą. Metoda wykorzystuje sąsiedztwo, aby dokonać klasyfikacji dotyczącej pojedynczego punktu danych. Model k najbliższych sąsiadów należy do rodziny modelów Lazy learning, co oznacza, że zachowuje w pamięci cały zbiór treningowy, który wykorzystuje do obliczeń podczas klasyfikacji. Z uwagi na to, że metoda w dużej mierze bazuje na pamięci, aby przechować dane treningowe, często mówi się, że jest to metoda memory based lub instance-based. W celu wyznaczenia najbliższych sąsiadów model potrzebuje miary, na podstawie której będzie oceniał odległość między punktami. Najczęściej wykorzystuje się odległość: euklidesową, miejską, Minkowskiego oraz Hamminga. Algorytm może być wykorzystywany do regresji, lecz wtedy prognoza jest liczona na podstawie średniej wartości k najbliższych sąsiadów [13].

3.6.6 Maszyny wektorów nośnych

Maszyny wektorów nośnych (ang. Support Vector Machines lub SVM) są rodziną metod uczenia maszynowego, które początkowo były wykorzystywane do rozwiązywania problemu klasyfikacji, jednakże później zostały one stosowane w różnych innych kontekstach. Metody opierają się na zasadach statystycznej teorii uczenia i

optymalizacji wypukłej w celu znalezienia odpowiedniej granicy w przestrzeni danych tym samym separując dwie klasy punktów [14].

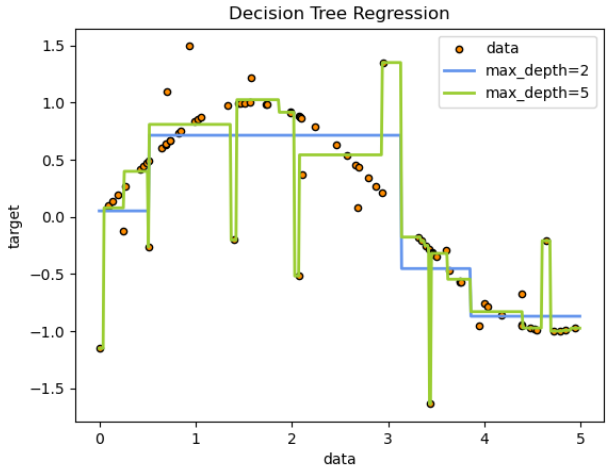
Support Vector Regression (SVR) to metoda będąca członkiem rodziny maszyn wektorów nośnych. Pozwala ona na rozwiązywanie problemów regresji. Celem algorytmu jest znalezienie hiperpłaszczyzny, która jest jak najlepiej dopasowana do punktów danych w przestrzeni. Proces ten polega na odwzorowaniu zmiennych wejściowych do przestrzeni cech o wyższych wymiarach i znalezieniu hiperpłaszczyzny, która maksymalizuje margines (odległość) między hiperpłaszczyzną a najbliższymi punktami danych, jednocześnie minimalizując błąd predykcji. SVR umożliwia odnajdywanie nieliniowych zależności między zmiennymi wejściowymi a zmienną docelową poprzez wykorzystanie funkcji jądra do przekształcenia danych do przestrzeni o wyższej wymiarowości [15].

Dokładna metoda wykorzystana w pracy to „Epsilon-Support Vector Regression", będąca wariantem metody SVR. Metoda pozwala na manipulacje parametrem ϵ, który określa szerokość marginesu tolerancji, dla którego błędy prognoz są akceptowalne i niekarane w trakcie procesu uczenia [6]. Ze względu na przynależność metody do rodziny maszyn wektorów nośnych, będzie ona w dalszej części pracy nazywana w ten sposób.

3.6.7 Drzewa decyzyjne

Drzewa decyzyjne to metoda uczenia nadzorowanego, wykorzystywana do klasyfikacji i regresji. Jest to metoda nieparametryczna, podobnie jak k najbliższych sąsiadów. Głównym celem tej metody jest tworzenie modelu, który prognozuje wartość zmiennej docelowej na podstawie zestawu reguł decyzyjnych, wydobywających cechy charakterystyczne z danych [16].

Jak można zaobserwować na rysunku 3.9, im głębsze jest drzewo, tym bardziej złożone są reguły decyzyjne. Możliwość klarownej wizualizacji oraz interpretacji drzew decyzyjnych stanowi istotną zaletę tej metody. Drzewa decyzyjne są modelami typu white box, co oznacza, że obserwacje w modelu mogą być dokładnie wyjaśnione za pomocą logiki. Model white box jest przeciwieństwem modelu black box, który charakteryzuje się trudnościami w interpretacji wyników, jak ma to miejsce w przypadku sieci neuronowych. Istotną wadą drzew decyzyjnych jest ich tendencja do tworzenia zbyt złożonych struktur podczas procesu uczenia, co prowadzi do ograniczonej zdolności modelu do poprawnego prognozowania na nowych danych [16].



Rysunek 3.9. Rysunek przedstawia reguły decyzyjne drzew decyzyjnych uczonych na danych zbliżonych do krzywej sinus w zależności od głębokości drzewa. Źródło: [16]

3.6.8 Lasy losowe

Lasy losowe (ang. Random Forests), stworzone przez Leo Breimana w 2001 roku, to algorytm, który odniósł wielki sukces jako metoda klasyfikacji ogólnej i regresji. Ta technika łączy wiele drzew decyzyjnych wygenerowanych losowo, a następnie agreguje ich prognozy poprzez uśrednianie. Dzięki temu podejściu metoda osiąga doskonałe wyniki, szczególnie w przypadkach, gdy liczba zmiennych jest znacznie większa niż liczba obserwacji [17]. Lasy losowe należą do rodziny metod Ensemble, które łączą prognozy wielu bazowych estymatorów zbudowanych przez określony algorytm w celu poprawy zdolności do trafnego prognozowania na nowych danych w porównaniu z pojedynczym estymatorem [18].

3.6.9 AdaBoost

AdaBoost to algorytm autorstwa Yoava Freund i Roberta Schapire. Jest to pierwszy praktyczny algorytm Boostingu, czyli podejścia w uczeniu maszynowym opartego na idei tworzenia dokładnej reguły prognozy poprzez łączenie wielu względnie słabych i niedokładnych reguł [19]. Podstawową zasadą działania algorytmu AdaBoost, jest dopasowywanie sekwencji słabych modeli uczenia, jak na przykład niewielkie drzewa decyzyjne, do wielokrotnie zmodyfikowanych wersji danych. Prognozy uzyskane z każdego z tych modeli są następnie łączone poprzez ważone głosowanie większościowe lub sumowanie w celu uzyskania ostatecznej prognozy. Modyfikacje danych w każdej tzw. iteracji boostingu polegają na przypisywaniu wag każdej próbce w

zbiorze treningowym. Początkowo wagi są przypisane w taki sposób, aby pierwszy krok polegał na szkoleniu słabego modelu na oryginalnych danych. W kolejnych iteracjach wagi próbek są indywiduwalnie modyfikowane, a algorytm uczenia jest ponownie stosowany do danych, na których przeprowadzono odpowiednie ważenie. W danym kroku wagi próbek, które zostały błędnie przewidziane przez model wzmocniony w poprzednim kroku, są zwiększane, podczas gdy wagi są zmniejszane dla próbek, które zostały poprawnie przewidziane. W miarę postępu iteracji, przykłady trudne do przewidzenia zyskują coraz większy wpływ. Każdy kolejny słaby model jest zatem zmuszony koncentrować się na przykładach pominiętych przez poprzednie modele w sekwencji [18].

3.6.10 Gradient boosting

Gradient boosting to technika uczenia maszynowego opracowana w pracy Jerome'a H. Friedmana w 2001 roku, która stanowi rozwinięcie idei boostingu. W przeciwieństwie do metody AdaBoost, która skupia się na dostosowywaniu wag próbek, gradient boosting opiera się na minimalizacji funkcji straty poprzez sekwencyjne budowanie modeli. W każdej iteracji nowy model jest trenowany tak, aby dopasować się do ujemnego gradientu wybranej funkcji straty, co efektywnie minimalizuje ogólny błąd predykcji. Proces ten opiera się na podejściu zachłannym, stopniowo poprawiającym wydajność modelu poprzez dodawanie prostych modeli bazowych, zwykle drzew decyzyjnych [21].

3.7 Optymalizacja hiperparametrów

Hiperparametry są parametrami, których wartości nie są wyznaczane na podstawie danych, a są deklarowane przed procesem trenowania modelu. Hiperparametry są odpowiedzialne za kontrolę procesu uczenia modelu i mają istotny wpływ na jego wydajność.

Każda z metod uczenia maszynowego wymienionych w poprzedniej sekcji, z wyłączeniem regresji liniowej, pozwala na dostosowanie hiperparametrów. W celu wyboru odpowiednich hiperparametrów dla modeli poszczególnych metod wykorzystano szukanie na siatce punktów (ang. grid-search) i kroswalidację (ang. cross-validation). Dla każdej kombinacji hiperparametrów ocena wydajności modelu jest dokonywana poprzez zastosowanie k-krotna kroswalidacja.

Szukanie na siatce punktów to technika optymalizacji hiperparametrów w uczeniu maszynowym, która przeszukuje zadeklarowaną siatkę hiperparametrów w celu znalezienia najlepszej kombinacji hiperparametrów pod kątem wydajności modelu [23].

K-krotna kroswalidacja (zwana również k-krotną walidacją krzyżową) jest techniką, która pomaga oszacować, jak model uczenia maszynowego będzie się sprawdzał na niezależych danych, nie wykorzystując przy tym zbioru testowego. Technika polega na podzieleniu zbioru treningowego na k mniejszych zbiorów. Następnie dany model jest trenowany na $k-1$ z k wydzielonych zbiorów i zostaje poddany ewaluacji na pozostałym wydzielonym zbiorze. Opisany proces trenowania i ewaluacji przeprowadzony zostaje k razy, przy czym każdy z wyznaczonych zbiorów zostaje wykorzystany do ewaluacji jedynie raz. Wydajność modelu jest oceniana na podstawie wybranej metryki, a otrzymane wyniki z kroswalidacji są uśredniane, co pozwala uzyskać bardziej obiektywną ocenę wydajności modelu dla danej konfiguracji hiperparametrów [24].

W pracy podczas procesu kroswalidacji metryką wykorzystaną do oceny wydajności modelu był błąd RMSE (Root Mean Square Error), czyli pierwiastek błędu średniokwadratowego, który zostanie dokładniej opisany w dalszej części pracy.

3.7.1 Specyfikacja siatek hiperparametrów

W tej podsekcji podane zostały siatki hiperparametrów wybrane dla poszczególnych metod wraz z wartościami i opisami hiperparametrów. Opisy hiperparametrów zostały oparte na dokumentacji biblioteki scikit-learn [6].

Regresja grzbietowa

- `alpha`: 0,001; 0,01; 0,1; 0,2; 0,3; 0,4; 0,5; 1; 10; 100

Hiperparametr `alpha` odpowiada symbolowi λ podanemu w sekcji opisującej regresję grzbietową. Hiperparametr kontroluje siłę nałożonej kary.

Lasso

- `alpha`: 0,001; 0,01; 0,1; 0,2; 0,3; 0,4; 0,5; 1; 10; 100

Hiperparametr `alpha` odpowiada symbolowi λ podanemu w sekcji opisującej metodę Lasso. Hiperparametr kontroluje siłę nałożonej kary.

Elastic net

- `alpha`: 0,001; 0,01; 0,1; 0,2; 0,3; 0,4; 0,5; 1; 10; 100
- `l1_ratio`: 0,1; 0,3; 0,5; 0,7; 0,9

Hiperparametr `alpha` odpowiada sumie symbolów λ_1 i λ_2 podanych w sekcji opisującej metodę elastic net. Hiperparametr `l1_ratio` przyjmuje wartości od 0 do 1 włącznie i reprezentuję jaką część parametru `alpha` stanowi λ_2.

K najbliższych sąsiadów

- `n_neighbors`: 3, 5, 7, 9, 11, 13, 15
- `weights`: uniform, distance
- `p`: 1, 2

Hiperparametr `n_neighbors` odpowiada k, a więc liczbie sąsiadów wykorzystywanych w metodzie. Hiperparametr `weights` określa jaki wpływ mają odległości sąsiadów podczas prognozowania. Gdy hiperparametr `weights` przyjmuje wartość „uniform", to sąsiedzi mają jednakowy wpływ, a gdy przyjmuje wartość „distance", to wpływ sąsiadów jest odwrotnie proporcjonalny do ich odległości. Hiperparametr `p` jest związany z odległością Minkowskiego. Gdy `p` przyjmuje wartość 1, odległość jest odległością miejską, a gdy `p` przyjmuje wartość 2, odległość jest odległością euklidesową.

Maszyny wektorów nośnych

- `kernel`: linear, rbf
- `C`: 0,1; 1; 10
- `gamma`: scale, auto
- `epsilon`: 0,1; 0,2; 0.5

Hiperparametr `kernel` dyktuje, jaki typ jądra zostanie wykorzystany w algorytmie. Hiperparametr `C` jest parametrem regularyzacji. Hiperparametr `gamma` dyktuje formę współczynnika w sytuacji, gdy typ jądra to rbf (radial basis function). Hiperparametr `epsilon` określa margines tolerancji, dla którego błędy prognoz są akceptowalne i niekarane w trakcie procesu uczenia.

Drzewa decyzyjne

- `max_depth`: None, 10, 20, 30, 40, 50

- `min_samples_split`: 5, 10, 15, 20, 25
- `min_samples_leaf`: 1, 2, 4, 8

Hiperparametr `max_depth` reprezentuje maksymalna możliwą głębokość drzewa decyzyjnego. Hiperparametr `min_samples_split` określa liczbę próbek w węźle, która jest wymagana do podzielenia węzła. Hiperparametr `min_samples_leaf` określa minimalną liczbę próbek, jaką liść musi zawierać, aby został utworzony.

Lasy losowe

- `n_estimators`: 10, 50, 100, 200
- `max_depth`: None, 10, 20, 30
- `min_samples_split`: 2, 5, 10
- `min_samples_leaf`: 1, 2, 4

Hiperparametry `max_depth`, `min_samples_split`, `min_samples_leaf` są to hiperparametry drzew decyzyjnych i zostały opisane w opisie hiperparametrów drzewa decyzyjnego. Hiperparametr `n_estimators` określa liczbę drzew decyzyjnych wykorzystanych w modelu lasów losowych.

AdaBoost

- `n_estimators`: 50, 100, 150
- `learning_rate`: 0,01; 0,1; 0,5; 1

Hiperparametr `n_estimators` określa liczbę słabych modeli uczenia wykorzystanych w metodzie, a konkretnie drzew decyzyjnych o maksymalnej głębokości równej 3. Hiperparametr `learning_rate` reprezentuje współczynnik uczenia, który kontroluje, jak bardzo każdy kolejny model przyczynia się do korekty błędów poprzednich modeli podczas procesu boostingu.

Gradient boosting

- `n_estimators`: 50, 100, 150
- `learning_rate`: 0,01; 0,1; 0,5; 1
- `max_depth`: 3, 4, 5

Hiperparametr `n_estimators` określa liczbę słabych modeli uczenia wykorzystanych w metodzie, a konkretnie drzew decyzyjnych o maksymalnej głębokości równej wartości hiperparametru `max_depth`. Hiperparametr `learning_rate` reprezentuje

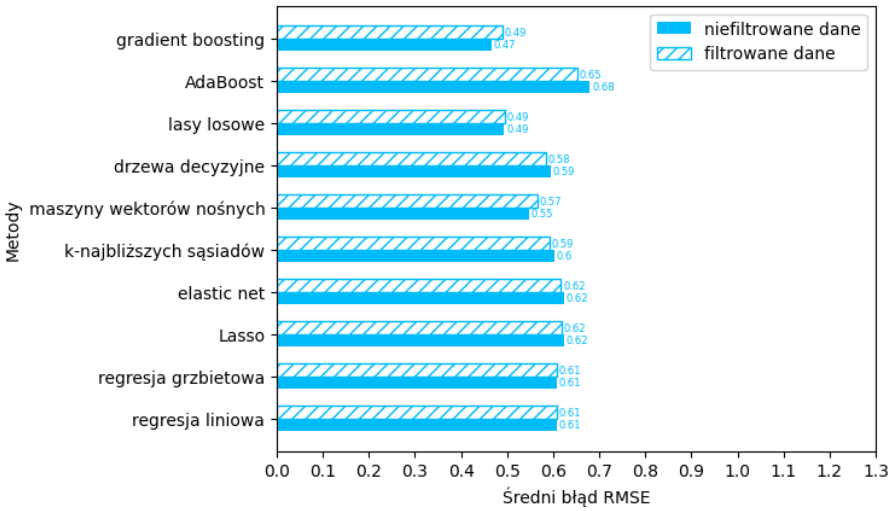
współczynnik uczenia, który zmniejsza wkład każdego drzewa o wartość tego hiperparametru.

3.7.2 Wstępna ocena modeli

Dla każdej z wybranych metod uczenia maszynowego wyselekcjonowano modele o optymalnych hiperparametrach, które w wyniku kroswalidacji zapewniają najlepsze wyniki, czyli minimalne średnie błędy RMSE. W tej sekcji przedstawione zostały wartości hiperparametrów modeli wraz z ich odpowiadającymi średnimi błędami RMSE. Wyniki zostały przedstawione na wykresach, aby umożliwić porównanie poszczególnych modeli oraz zobaczyć, jakie wyniki osiągają modele w zależności od charakterystyki danych. Pomimo tego, że regresja liniowa nie pozwala na manipulowanie hiperparametrami, to wyniki jej kroswalidacji zostały również przedstawione w celu porównania z wynikami innych modeli.

Tabela 3.3. Tabela przedstawia optymalne wartości hiperparametrów dla poszczególnych metod, uzyskane w wyniku kroswalidacji na niefiltrowanym oraz filtrowanym zbiorze treningowym „California Housing". Opracowanie własne.

Metoda	**Hiperparametry dla niefiltrowanych danych**	**Hiperparametry dla filtrowanych danych**
Regresja grzbietowa	`alpha`: 0,001	`alpha`: 0,001
Lasso	`alpha`: 0,001	`alpha`: 0,001
Elastic net	`alpha`: 0,001 `l1_ratio`: 0,9	`alpha`: 0,001 `l1_ratio`: 0,1
K najbliższych sąsiadów	`n_neighbors`: 15 `p`: 1 `weights`: distance	`n_neighbors`: 15 `p`: 1 `weights`: distance
Maszyny wektorów nośnych	`C`: 10 `epsilon`: 0,2 `gamma`: scale `kernel`: rbf	`C`: 1 `epsilon`: 0,2 `gamma`: scale `kernel`: rbf
Drzewa decyzyjne	`max_depth`: 10 `min_samples_leaf`: 8 `min_samples_split`: 25	`max_depth`: 10 `min_samples_leaf`: 8 `min_samples_split`: 25
Lasy losowe	`max_depth`: 30 `min_samples_leaf`: 2 `min_samples_split`: 5 `n_estimators`: 200	`max_depth`: 20 `min_samples_leaf`: 1 `min_samples_split`: 2 `n_estimators`: 200
AdaBoost	`learning_rate`: 0,1 `n_estimators`: 100	`learning_rate`: 0,1 `n_estimators`: 50
Gradient boosting	`learning_rate`: 0,1 `max_depth`: 5 `n_estimators`: 150	`learning_rate`: 0,1 `max_depth`: 5 `n_estimators`: 150



Rysunek 3.10. Wykres przedstawia minimalne średnie błędy RMSE osiągnięte dla poszczególnych metod uczenia maszynowego w wyniku procesu szukania na siatce punktów i kroswalidacji na zbiorze treningowym „California Housing" oraz jego filtrowanej wersji. Opracowanie własne.

Na rysunku 3.10 zauważalne jest, że zastosowane metody, które tworzą modele uczące się na zbiorze treningowym „California Housing" oraz jego przefiltrowanej wersji, osiągają bardzo zbliżone wyniki. Warto zaznaczyć, że między wynikami poszczególnych metod nie występują znaczące różnice. Poza lasami losowymi i gradient boosting, dla których udało się uzyskać średni błąd RMSE poniżej 0,5, pozostałe metody wykazują średnie błędy RMSE w granicach 0,6. Interesującym zjawiskiem jest fakt, że metoda AdaBoost uzyskała gorsze wyniki niż metody liniowe (regresja liniowa, regresja grzbietowa, Lasso, elastic net), pomimo swojej większej złożoności. Dla zbioru „California Housing" oraz jego przefiltrowanej wersji trzy metody, których modele osiągnęły najlepsze wyniki, to gradient boosting, lasy losowe oraz maszyny wektorów nośnych.

Analizując rysunek 3.11, natychmiastowo rzuca się w oczy, że modele, które zostały poddane treningowi na filtrowanym zbiorze treningowym „Miami Housing", uzyskują wyraźnie lepsze wyniki w kroswalidacji w porównaniu do odpowiadających im modeli trenowanych na danych niefiltrowanych. Ta tendencja jest zauważalna we wszystkich badanych metodach. Podobnie jak w przypadku zbiorów „California Housing" (por. rysunek 3.10), dla wariantów zbiorów „Miami Housing" metody, które wyraźnie wyprzedzają pozostałe, to gradient boosting i lasy losowe. Ciekawą obserwacją jest fakt, że metoda maszyn wektorów nośnych osiągnęła najgorsze wyniki zarówno dla danych niefiltrowanych, jak i filtrowanych, pomimo że zajmowała trzecie miejsce pod względem efektywności dla wariantów zbiorów „California Housing"

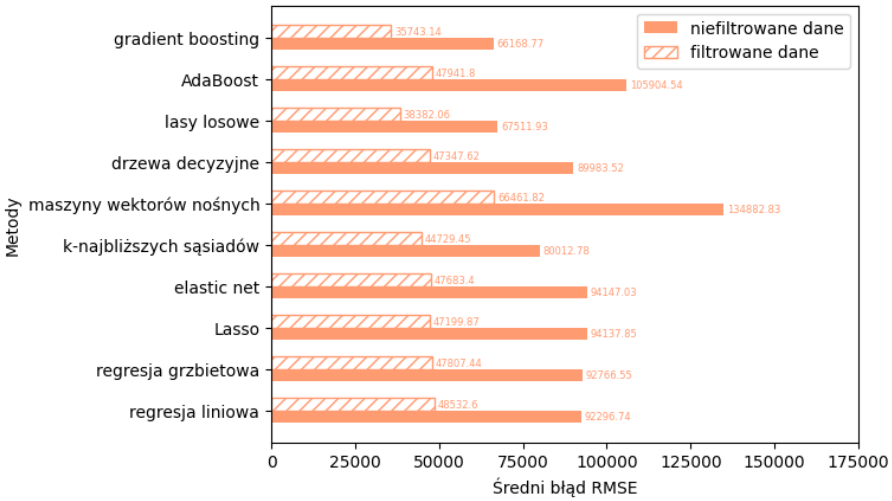
Tabela 3.4. Tabela przedstawia optymalne wartości hiperparametrów dla poszczególnych metod, uzyskane w wyniku kroswalidacji na niefiltrowanym oraz filtrowanym zbiorze treningowym „Miami Housing". Opracowanie własne.

Metoda	**Hiperparametry dla niefiltrowanych danych**	**Hiperparametry dla filtrowanych danych**
Regresja grzbietowa	`alpha`: 0,001	`alpha`: 10
Lasso	`alpha`: 10	`alpha`: 100
Elastic net	`alpha`: 0,001 `l1_ratio`: 0,7	`alpha`: 0,1 `l1_ratio`: 0,9
K najbliższych sąsiadów	`n_neighbors`: 7 `p`: 1 `weights`: distance	`n_neighbors`: 5 `p`: 1 `weights`: distance
Maszyny wektorów nośnych	`C`: 10 `epsilon`: 0,1 `gamma`: scale `kernel`: linear	`C`: 10 `epsilon`: 0,5 `gamma`: scale `kernel`: linear
Drzewa decyzyjne	`max_depth`: None `min_samples_leaf`: 8 `min_samples_split`: 5	`max_depth`: 10 `min_samples_leaf`: 4 `min_samples_split`: 20
Lasy losowe	`max_depth`: 20 `min_samples_leaf`: 2 `min_samples_split`: 5 `n_estimators`: 200	`max_depth`: 30 `min_samples_leaf`: 2 `min_samples_split`: 2 `n_estimators`: 200
AdaBoost	`learning_rate`: 0,1 `n_estimators`: 150	`learning_rate`: 0,5 `n_estimators`: 150
Gradient boosting	`learning_rate`: 0,1 `max_depth`: 5 `n_estimators`: 150	`learning_rate`: 0,1 `max_depth`: 4 `n_estimators`: 150

(por. rysunek 3.10). Dla zbioru „Miami Housing" oraz jego przefiltrowanej wersji trzy metody, których modele uzyskały najlepsze wyniki, to gradient boosting, lasy losowe oraz k najbliższych sąsiadów.

3.8 Trenowanie modeli

W poprzedniej sekcji dokonano wstępnej oceny wybranych metod uczenia maszynowego na różnych wariantach zbiorów treningowych przy użyciu kroswalidacji. Dla wariantów zbioru treningowego „California Housing" najlepsze wyniki osiągnęły modele: gradient boosting, lasy losowe oraz maszyny wektorów nośnych. Natomiast, dla wariantów zbioru treningowego „Miami Housing" najlepsze rezultaty uzyskały modele: gradient boosting, lasy losowe oraz k najbliższych sąsiadów. Ze względu na ich wysoką skuteczność, te modele zostały wybrane do dalszego treningu na całkowitych danych treningowych. Celem tego działania jest stworzenie jak najdokładniejszych



Rysunek 3.11. Wykres przedstawia minimalne średnie błędy RMSE osiągnięte dla poszczególnych metod uczenia maszynowego w wyniku procesu szukania na siatce punktów i kroswalidacji na zbiorze treningowym „Miami Housing" oraz jego filtrowanej wersji. Opracowanie własne.

modeli, które będą skutecznie prognozować ceny nieruchomości na podstawie nowych danych.

3.9 Ewaluacja modeli

W celu ewaluacji modeli wykorzystane zostały trzy metryki. Pierwszą z wykorzystanych metryk jest błąd RMSE (Root Mean Square Error), a więc pierwiastek błędu średniokwadratowego, który można zapisać w następujący sposób:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.7}$$

, gdzie:

- n - liczba obserwacji
- y_i - wartość rzeczywista
- $\hat{y}_i$ - wartość prognozowana

Źródło: [25]

Kolejną z wykorzystanych metryk jest błąd względny, który można zapisać w następujący sposób:

$$\delta = \frac{|y_i - \hat{y}_i|}{y_i} \tag{3.8}$$

, gdzie:

- y_i - wartość rzeczywista
- $\hat{y}_i$ - wartość prognozowana

Źródło: [26]

Ostatnią z metryk wykorzystanych do ewaluacji modeli jest błąd MAPE (Mean Absolute Percentage Error), który można zapisać w sposób:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{y_i}| \tag{3.9}$$

, gdzie:

- n - liczba obserwacji
- y_i - wartość rzeczywista
- $\hat{y}_i$ - wartość prognozowana

Źródło: [25]

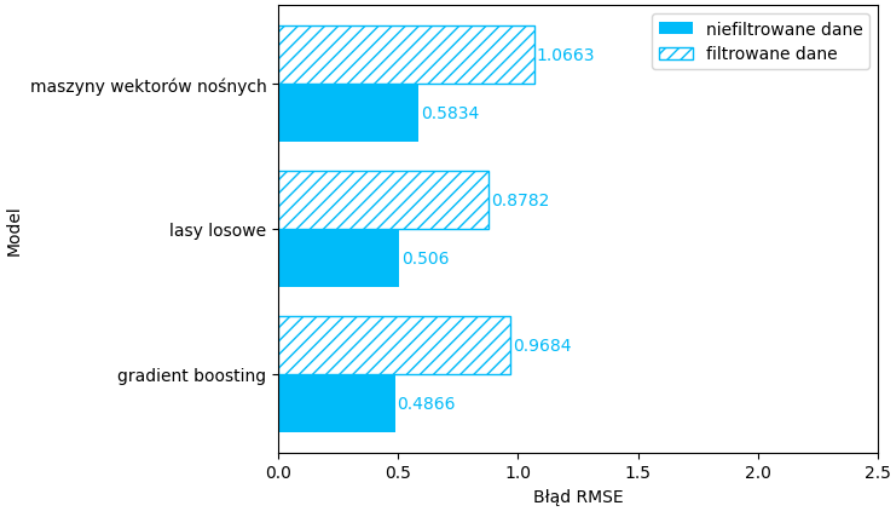
4 Omówienie Wyników

W niniejszym rozdziale zaprezentowano, zwizualizowano oraz opisano wyniki ewaluacji wybranych modeli.

Tabela 4.1. Tabela przedstawia wyniki ewaluacji modeli na zbiorze testowym w zależności od użytego wariantu zbioru treningowego. Opracowanie własne. Wartości błędów zostały zaokrąglone do czterech miejsc po przecinku.

Model	**Zbiór treningowy**	**RMSE**	**Średni błąd względny**	**MAPE**
gradient boosting	„California Housing" (niefiltrowany)	0,4866	0,1874	18,7354%
lasy losowe	„California Housing" (niefiltrowany)	0,5060	0,1850	18,4979%
maszyny wektorów nośnych	„California Housing" (niefiltrowany)	0,5834	0,2302	23,0244%
gradient boosting	„California Housing" (filtrowany)	0,9684	0,5049	50,4870%
lasy losowe	„California Housing" (filtrowany)	0,8782	0,4420	44,2043%
maszyny wektorów nośnych	„California Housing" (filtrowany)	1,0663	0,5137	51,3735%
gradient boosting	„Miami Housing" (niefiltrowany)	120680,6682	0,1131	11,3134%
lasy losowe	„Miami Housing" (niefiltrowany)	132199,1016	0,1132	11,3232%
k-najbliższych sąsiadów	„Miami Housing" (niefiltrowany)	172969,8302	0,1367	13,6725%
gradient boosting	„Miami Housing" (filtrowany)	200111,5442	0,4217	42,1654%
lasy losowe	„Miami Housing" (filtrowany)	193956,4843	0,3492	34,9223%
k-najbliższych sąsiadów	„Miami Housing" (filtrowany)	189823,5482	0,2412	24,1224%

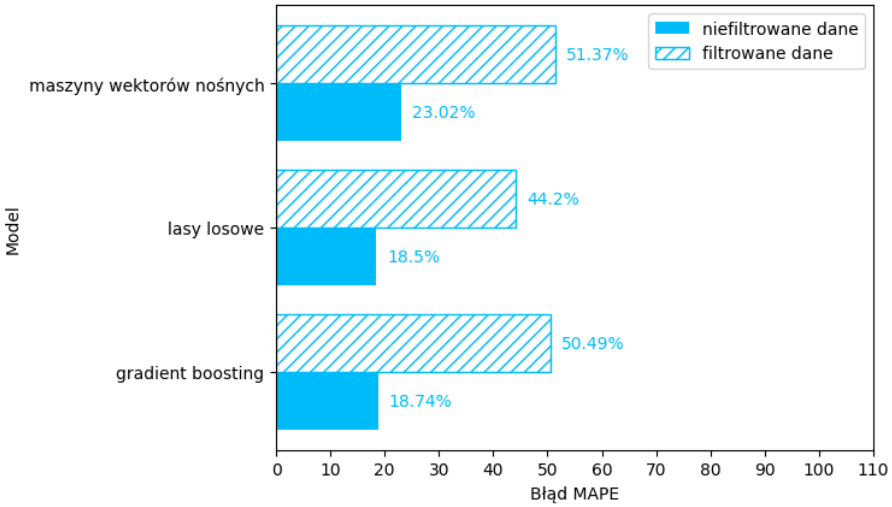
Analizując wykres przedstawiony na rysunku 4.2, można zauważyć, że model lasów losowych trenowany na niefiltrowanych danych osiągnął najniższy błąd MAPE na poziomie $18,5\%$. Nieco gorszy wynik, $18,74\%$, uzyskał model gradient boosting trenowany na danych niefiltrowanych. Spośród trzech modeli trenowanych na niefiltrowanych danych, najwyższy błąd MAPE, wynoszący $23,02\%$, osiągnął model maszyn wektorów nośnych. W przypadku modeli trenowanych na filtrowanych danych, najwyższy błąd MAPE również został odnotowany dla modelu maszyn wektorów nośnych, wynoszący $51,37\%$. Niewiele niższy wynik, $50,49\%$, uzyskał model



Rysunek 4.1. Na wykresie zaprezentowano błędy RMSE modeli, które zostały wytrenowane na różnych wariantach zbioru „California Housing" i poddane ewaluacji. Opracowanie własne na podstawie tabeli 4.1. Błędy zostały zaokrąglone do czterech miejsc po przecinku.

gradient boosting trenowany na filtrowanych danych. Najlepszy rezultat w tej grupie osiągnął model lasów losowych, którego błąd MAPE wyniósł $44,2\%$. Mimo że jest to wynik lepszy niż dla dwóch pozostałych modeli trenowanych na filtrowanych danych, wciąż znacząco przewyższa on wyniki modeli trenowanych na danych niefiltrowanych. Analiza wykresu na rysunku 3.10 pokazała, że w procesie wstępnej oceny metod nie zaobserwowano wyraźnego wpływu wariantu zbioru treningowego „California Housing" na średnie błędy RMSE modeli. Jednakże na wykresie na rysunku 4.1 widać, że ta zależność nie miała miejsca w przypadku ewaluacji na zbiorze testowym. Modele poszczególnych metod osiągały wyraźnie lepsze wyniki, gdy były trenowane na danych niefiltrowanych. Błędy MAPE modeli trenowanych na filtrowanych danych były ponad dwukrotnie większe niż błędy modeli tych samych metod trenowanych na danych niefiltrowanych.

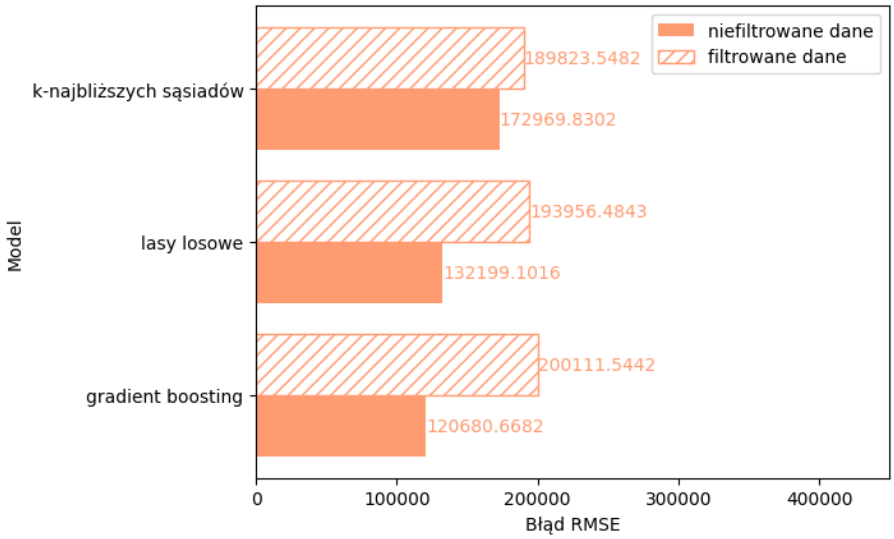
Na wykresie przedstawionym na rysunku 4.4 można zauważyć, że podczas ewaluacji na zbiorze testowym modele trenowane na danych niefiltrowanych osiągały wyraźnie mniejsze błędy MAPE niż modele trenowane na filtrowanych danych. Na wykresie przedstawionym na rysunku 3.11 widać było, że podczas procesu wstępnej oceny modeli, osiągały one znacznie mniejsze średnie błędy RMSE, gdy były trenowane na danych filtrowanych. Patrząc na wykres przedstawiony na rysunku 4.3 jest widoczne, że ta zależność nie ma miejsca podczas ewaluacji modeli na zbiorze testowym. Można powiedzieć, że istnieje przeciwna zależność, a mianowicie, że modele trenowane na danych filtrowanych osiągały znacznie większe błędy RMSE



Rysunek 4.2. Na wykresie zaprezentowano błędy MAPE modeli, które zostały wytrenowane na różnych wariantach zbioru „California Housing" i poddane ewaluacji. Opracowanie własne na podstawie tabeli 4.1. Błędy zostały zaokrąglone do dwóch miejsc po przecinku.

w porównaniu modeli trenowanych na danych niefiltrowanych. Na wykresie przedstawionym na rysunku 4.2 zauważalne jest, że najmniejszy błąd MAPE (11,31%) osiągnął model gradient boosting trenowany na danych niefiltrowanych. Modelem, który uzyskał minimalnie większy błąd MAPE (11,32%) jest model lasów losowych trenowany na niefiltrowanych danych. Modelem trzecim w kolejności, jeśli chodzi o najmniejszy błąd MAPE podczas ewaluacji, jest model k najbliższych sąsiadów trenowany na danych niefiltrowanych. Model ten osiągnał błąd MAPE niewiele większy od dwóch wcześniej omawianych modeli, bo wynoszący 13,67%. Kolejnym modelem jest model k najbliższych sąsiadów trenowany na filtrowanych danych, który uzyskał błąd MAPE wynoszący 24,12%. Wynik wyraźnie gorszy (34,92%) osiągnął model lasów losowych trenowany na filtrowanych danych. Największy błąd MAPE spośród wybranych modeli, wynoszący 42,17%, uzyskał model gradient boosting trenowany na filtrowanych danych.

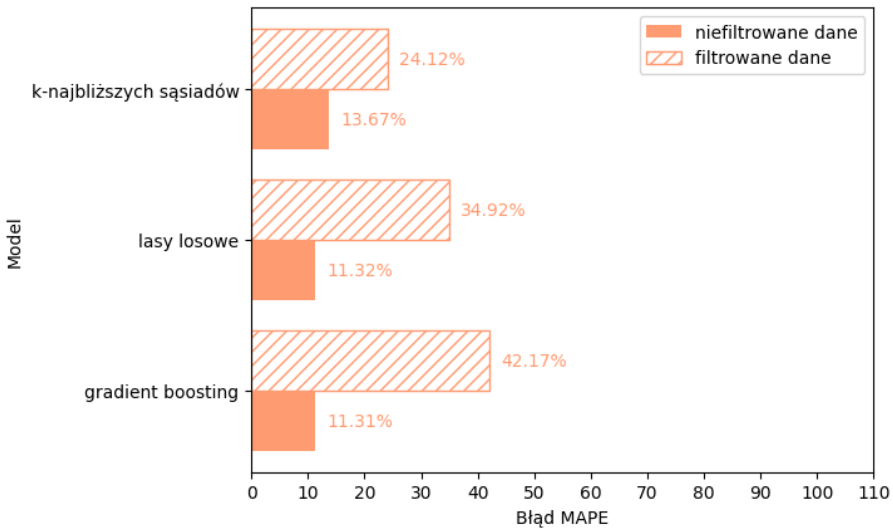
Na rysunku 4.5 zauważalne jest, że punkty modeli trenowanych na niefiltrowanych danych „California Housing" są umieszczone wyraźnie węziej przekątnej linii. Oznacza to, że prognozy tych modeli wyraźnie mniej odbiegały od rzeczywistych wartości niż prognozy modeli trenowanych na filtrowanych danych. Można również zauważyć na wykresach modeli trenowanych na filtrowanych danych, że większość punktów na tych wykresach jest położona nad przekątną linią. Oznacza to, że modele te mają tendencję do prognozowania zawyżonych cen nieruchomości. Warto również zauważyć, że modele napotykają szczególne trudności w przewidywaniu wartości



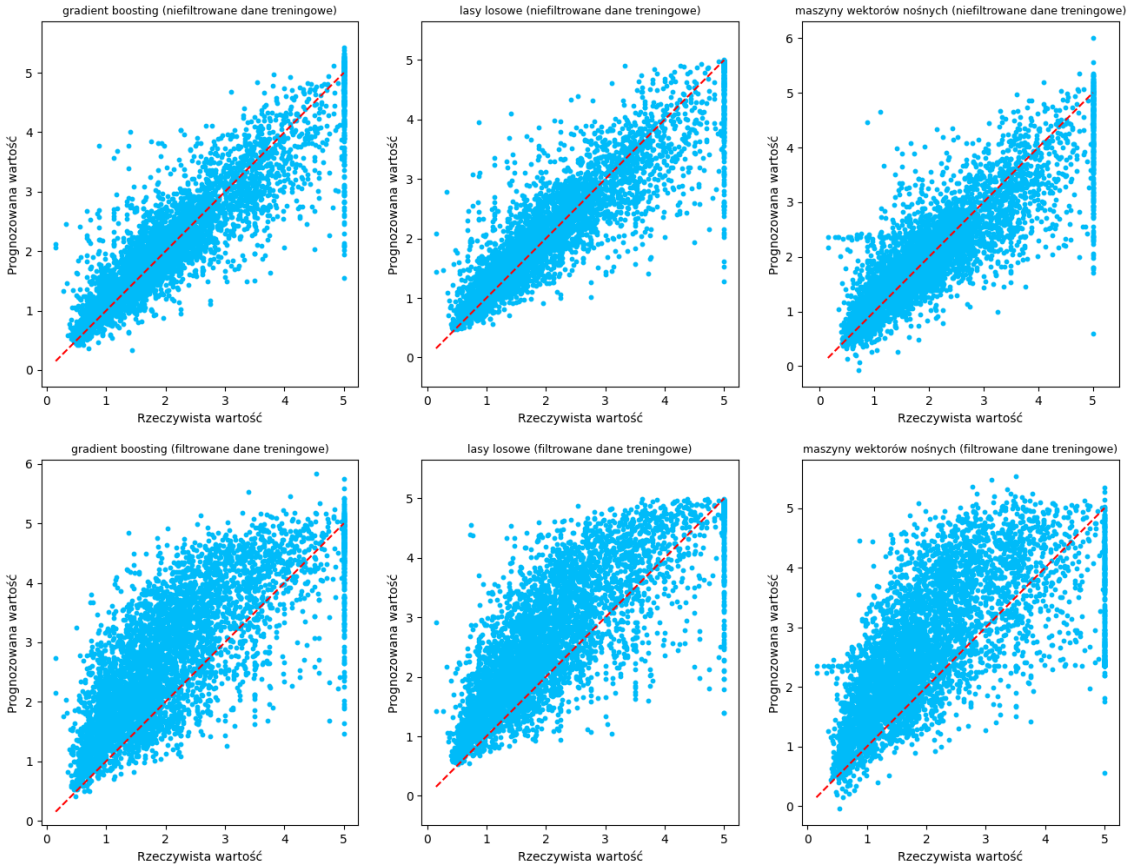
Rysunek 4.3. Na wykresie zaprezentowano błędy RMSE modeli, które zostały wytrenowane na różnych wariantach zbioru „Miami Housing" i poddane ewaluacji. Opracowanie własne na podstawie tabeli 4.1. Błędy zostały zaokrąglone do czterech miejsc po przecinku.

dla obserwacji reprezentujących nieruchomości w grupie bloków, których mediana wartości wynosi pół miliona dolarów. To zjawisko może być skutkiem wcześniej omawianego potencjalnego procesu formatowania danych, w którym mediana wartości nieruchomości w grupie bloków jest zaokrąglana do wartości pół miliona dolarów, gdy przekracza ten poziom.

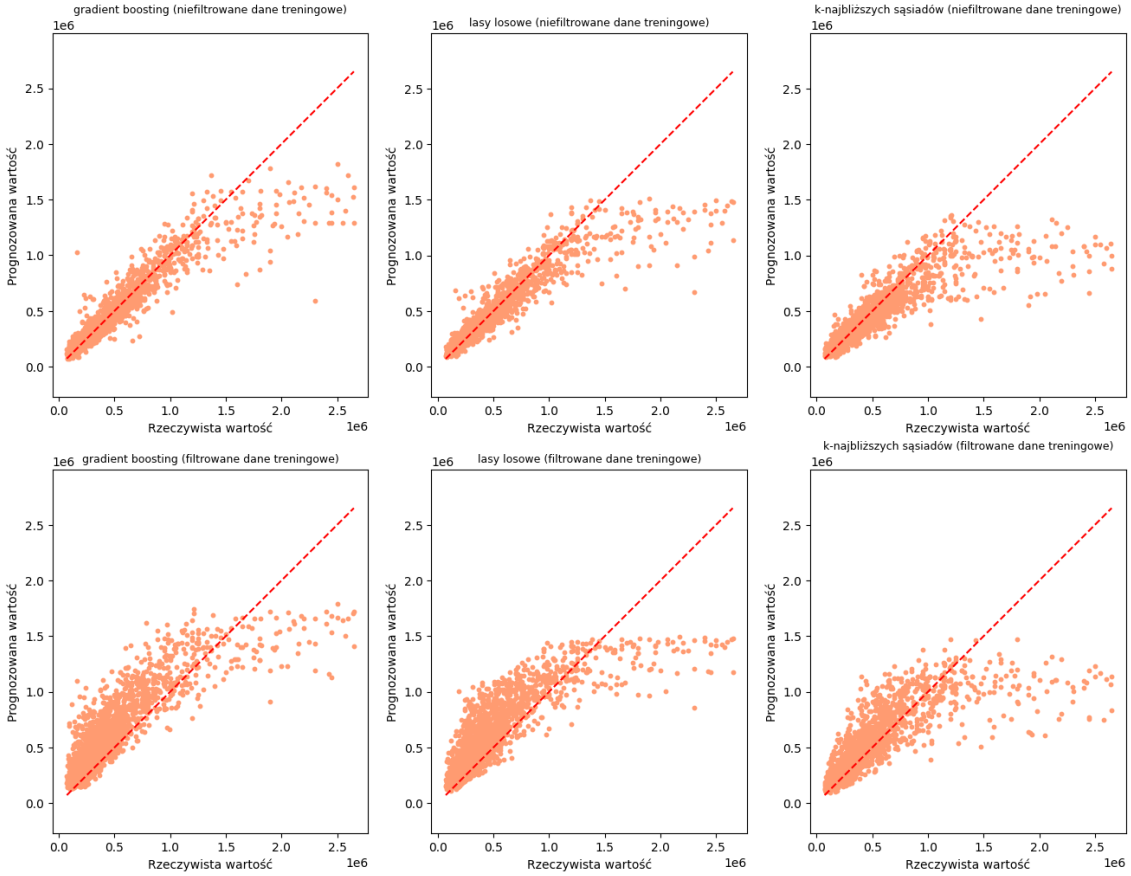
Patrząc na rysunek 4.6 dostrzegalne jest, że wszystkim modelom trenowanym na wariantach zbioru treningowego „Miami Housing" trudność sprawia prognoza cen nieruchomości o wysokich wartościach. Objawia się to dużą ilością punktów umieszczonych pod linią przekątną dla rzeczywistych wartości nieruchomości większych niż 1 500 000 dolarów. Dla nieruchomości o wartości rzeczywistej przekraczającej ten próg modele wydają się prognozować jedynie zaniżone wartości. Dla nieruchomości o wartości rzeczywistej poniżej omawianego progu, modele trenowane na filtrowanym zbiorze treningowym „California Housing" prognozują wartości znacznie bardziej odbiegające od rzeczywistych w porównaniu do prognoz modeli trenowanych na zbiorze treningowym niefiltrowanym. Dodatkowo, prognozy te wykazują tendencję do bycia zawyżonymi, podczas gdy prognozy modeli trenowanych na zbiorze treningowym niefiltrowanym są znacznie bardziej zrównoważone.



Rysunek 4.4. Na wykresie zaprezentowano błędy MAPE modeli, które zostały wytrenowane na różnych wariantach zbioru „Miami Housing" i poddane ewaluacji. Opracowanie własne na podstawie tabeli 4.1. Błędy zostały zaokrąglone do dwóch miejsc po przecinku.



Rysunek 4.5. Wykresy przedstawiają porównanie między rzeczywistymi wartościami zbioru testowego, a przewidywanymi przez modele, które były trenowane na niefiltrowanym lub filtrowanym zbiorze treningowym „California Housing". Punkty na wykresach reprezentują pojedyncze obserwacje. Linia przerywana reprezentuje idealne przypadki, w których rzeczywiste wartości są równe wartościom przewidywanym przez model. Opracowanie własne.



Rysunek 4.6. Wykresy przedstawiają porównanie między rzeczywistymi wartościami zbioru testowego, a przewidywanymi przez modele, które były trenowane na niefiltrowanym lub filtrowanym zbiorze treningowym „Miami Housing". Punkty na wykresach reprezentują pojedyncze obserwacje. Linia przerywana reprezentuje idealne przypadki, w których rzeczywiste wartości są równe wartościom przewidywanym przez model. Opracowanie własne.

5 Wnioski

Modele trenowane na filtrowanych wariantach zbiorów treningowych „California Housing" i „Miami Housing" uzyskiwały podczas ewaluacji na zbiorach testowych wyraźnie gorsze wyniki od modeli trenowanych na niefiltrowanych wariantach tych zbiorów. W związku z tym cel filtracji, którym było dopasowanie modeli do rzeczywistych warunków rynku nieruchomości, nie został osiągnięty. Wygląda na to, że modele trenowane na danych poddanych procesowi filtracji mogą wykazywać symptomy przetrenowania. Przetrenowanie jest to zjawisko, w którym model precyzyjnie odzwierciedla wzorce występujące w danych treningowych, ale ma ograniczoną zdolność do dokładnego przewidywania wartości dla nowych danych spoza zbioru treningowego [9]. Modele trenowane na danych poddanych filtracji mogą wykazywać gorsze wyniki z kilku przyczyn. Po pierwsze, proces filtracji może prowadzić do utraty istotnych informacji lub nadmiernego ograniczenia zakresu danych treningowych. Dodatkowo, dobór kryteriów filtracji, który miał ulepszyć modelowanie rynku nieruchomości, może wprowadzić zmiany w rozkładzie danych treningowych, co sprawia, że te dane nieodpowiednio odzwierciedlają rzeczywisty rozkład danych nieruchomości, co w efekcie może doprowadzić do przeciwnego efektu, niż oczekiwano.

Dla zbioru „California Housing" model lasów losowych trenowany na niefiltrowanych danych osiągnął najlepsze wyniki podczas ewaluacji, natomiast dla zbioru „Miami Housing" najlepsze wyniki uzyskał model gradient boosting trenowany na niefiltrowanych danych. Warto zauważyć, że modele lasów losowych i gradient boosting osiągnęły wyniki bardzo zbliżone. Dla zbioru „California Housing" model gradient boosting osiągnął błąd MAPE o $0,24$ punktu procentowego większy niż model lasów losowych, a dla zbioru „Miami Housing" model gradient boosting osiągnął błąd MAPE jedynie o $0,01$ punktu procentowego mniejszy niż model lasów losowych. Najniższe uzyskane błędy MAPE dla poszczególnych zestawów danych (dla „California Housing" - $18,5\%$, a dla „Miami Housing" - $11,3\%$) można uznać za zadowalające w kontekście skuteczności modelu. Wyniki sugerują, że modele są w stanie dokonywać prognoz cen nieruchomości z relatywnie niskim poziomem błędu w porównaniu z rzeczywistymi wartościami. Porównując te wyniki, zauważalne jest,

jak charakterystyka danych wpływa na ostateczne wyniki modeli, co objawia się tym, że najniższy błąd MAPE osiągnięty przez model trenowany na zbiorze „California Housing" jest półtora raza większy od najniższego błędu MAPE osiągniętego przez model trenowany na zbiorze „Miami Housing".

Warto zaznaczyć, że o ile błędy RMSE czy MAPE, stanowią istotny wskaźnik oceny efektywności modeli, ich interpretacja musi być uwarunkowana kontekstem biznesowym oraz oczekiwaniami klientów.

Istnieje wiele strategii, które mogą potencjalnie zwiększyć skuteczność modeli predykcyjnych. Po pierwsze, można rozważyć pozyskanie danych o wyższej jakości, co oznacza dane dokładne i zróżnicowane. Po drugie, istotną rolę mogą odgrywać różne techniki inżynierii cech, inne niż te, które zostały wykorzystane w pracy. Po trzecie, można rozważyć inny sposób obsługiwania obserwacji odstających w zbiorze treningowym. Po czwarte, można, zmienić kryterium filtracji na inne niż te, które zostało wykorzystane w badaniu. Ponadto, możliwe jest także zastosowanie alternatywnych metod uczenia maszynowego lub nawet sieci neuronowych.

6 Bibliografia

[1] *SGGW-Thesis-LaTeX* https://github.com/lchmiel/SGGW-Thesis-LaTeX (dostęp 22.02.2024)

[2] *Yu, Hujia, and Jiafu Wu. "Real estate price prediction with regression and classification." CS229 (Machine Learning) Final Project Reports (2016): 1-5.*

[3] *Truong, Quang, et al. "Housing price prediction via improved machine learning techniques." Procedia Computer Science 174 (2020): 433-442.*

[4] *Zhang, Yuanheng. Application of Machine Learning in Boston House Price Prediction. Advances in Economics, Management and Political Sciences. 43. (2023): 176-184.*

[5] *Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." 2008 eighth ieee international conference on data mining. IEEE, 2008: 413-422.*

[6] *scikit-learn Machine Learning in Python* https://scikit-learn.org/stable/index.html (dostęp 07.01.2024)

[7] *California Housing* https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html (dostęp 21.02.2024)

[8] *MiamiHousing2016* https://www.openml.org/search?type=data&sort=runs&id=43093&status=active (dostęp 21.02.2024)

[9] *Raschka, Sebastian, and Vahid Mirjalili. Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd, 2019.*

[10] *What is Feature Engineering — Importance, Tools and Techniques for Machine Learning* https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10 (dostęp 07.01.2024)

[11] *Importance of Feature Scaling* https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html (dostęp 21.01.2024)

[12] *Z-Score Normalization: Definition & Examples* https://www.statology.org/z-score-normalization/ (dostęp 21.02.2024)

[13] *What is the k-nearest neighbors algorithm?* https://www.ibm.com/topics/knn (dostęp 21.01.2024)

[14] *Mammone, Alessia, Marco Turchi, and Nello Cristianini. "Support vector machines." Wiley Interdisciplinary Reviews: Computational Statistics 1.3 (2009): 283-289.*

[15] *Support Vector Regression Tutorial for Machine Learning* https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/ (dostęp 29.02.2024)

[16] *Decision Trees* https://scikit-learn.org/stable/modules/tree.html#decision-trees (dostęp 21.01.2024)

[17] *Biau, Gérard, and Erwan Scornet. "A random forest guided tour." Test 25 (2016): 197-227.*

[18] *Ensembles: Gradient boosting, random forests, bagging, voting, stacking* https://scikit-learn.org/stable/modules/ensemble.html (dostęp 22.01.2024)

[19] *Schapire, Robert E. "Explaining adaboost." Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. 37-52.*

[20] *Kuhn, Max, and Kjell Johnson. Applied predictive modeling. Vol. 26. New York: Springer, 2013.*

[21] *Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.*

[22] *Kendall tau metric* https://encyclopediaofmath.org/index.php?title=Kendall_tau_metric (dostęp 31.01.2024)

[23] *Tuning the hyper-parameters of an estimator* https://scikit-learn.org/stable/modules/grid_search.html#grid-search (dostęp 03.01.2024)

[24] *Cross-validation: evaluating estimator performance* https://scikit-learn.org/stable/modules/cross_validation.html (dostęp 03.01.2024)

[25] *MSE vs RMSE vs MAE vs MAPE vs R-Squared: When to Use?* https://vitalflux.com/mse-vs-rmse-vs-mae-vs-mape-vs-r-squared-when-to-use/ (dostęp 20.02.2024)

[26] *Błąd bezwzględny i względny pomiaru* https://www.matemaks.pl/blad-bezwzgledny-i-wzgledny-pomiaru.html (dostęp 20.02.2024)

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym w Archiwum Prac Dyplomowych SGGW.

.....
(czytelny podpis autora pracy)