Systemy z uczeniem maszynowym

Zaawansowane przygotowanie danych w uczeniu maszynowym

PREDYKCJA CEN SAMOCHODÓW

RAPORT Z PROJEKTU

Szymon Cyperski 180408

1. OPIS SYSTEMU

1.1. Definicja problemu

System będzie przewidywał ceny samochodów na rynku polskim. Predykcje będą dokonywane na podstawie podstawowych informacji o pojeździe, które są powszechnie wykorzystywane podczas transakcji kupna lub sprzedaży. Działanie systemu będzie ograniczone wyłącznie do nieuszkodzonych samochodów osobowych.

Opisywany system będzie stanowił pomoc dla osób bez szerszej wiedzy na temat rynku samochodowego, które chcą jak najkorzystniej sprzedać lub kupić pojazd. System pozwoliłby im na zaoszczędzenie czasu, który zazwyczaj muszą one przeznaczyć na znalezienie podobnych egzemplarzy na stronach aukcyjnych, aby móc porównać ceny.

1.2. Wejście systemu

Zbiór danych zostanie zebrany za pomocą dedykowanego programu do automatycznego pozyskiwania informacji z popularnej strony **Otomoto.pl**, która zawiera oferty sprzedaży samochodów. Surowe dane pobrane ze strony zostaną następnie przetworzone oraz wykorzystane na etapie treningu oraz testów modelu. Do najczęstszych informacji, które można znaleźć w ogłoszeniach należą następujące pola:

- Marka, model, generacja oraz wersja
- Rok produkcji oraz przebieg
- Pojemność oraz moc silnika, rodzaj paliwa
- Rodzaj napędu oraz skrzyni biegów
- Typ nadwozia, liczba drzwi oraz liczba miejsc
- Kraj pochodzenia
- Stan pojazdu (nowy/używany/uszkodzony)
- Wyposażenie dodatkowe
- Lokalizacja oferty
- Opis ogłoszenia.

Docelowy zbiór cech wejściowych systemu zostanie wytypowany na etapie analizy i eksploracji danych otrzymanych w procesie wstępnego przetwarzania. Wśród nich najpewniej znajdą się cechy zarówno numeryczne, jak i tekstowe.

1.3. Wyjście systemu

System będzie zwracał nieujemną wartość liczbową odpowiadającą estymowanej wartości pojazdu wyrażonej w PLN. Jako etykiety bazowe na etapie treningu oraz testów modelu zostaną wykorzystane ceny pojazdów zawarte w ofertach sprzedaży. Zostaną one zebrane razem z pozostałymi informacjami o pojazdach w trakcie etapu gromadzenia danych.

2. PRZYGOTOWANIE DANYCH

2.1. Opis wymaganego zbioru danych

Do realizacji projektu potrzebny będzie zbiór danych z samochodami, który zawiera informacje o wartości każdego pojazdu oraz informacje o podstawowych cechach opisujących pojazd. Do zbioru najważniejszych cech należą między innymi następujące pola:

- Marka, model, generacja oraz wersja
- Rok produkcji oraz przebieg
- Pojemność oraz moc silnika, rodzaj paliwa
- Rodzaj napędu oraz skrzyni biegów
- Typ nadwozia, liczba drzwi oraz liczba miejsc
- Kraj pochodzenia
- Stan pojazdu (nowy/używany/uszkodzony)
- Wyposażenie dodatkowe.

Preferowaną formą reprezentacji zbioru jest postać tabelaryczna, gdzie każdy wiersz reprezentuje inny pojazd, a każda kolumna to inna cecha go opisująca. Ceny pojazdów powinny być zgodne z ich rzeczywistą wartością obowiązującą na rynku samochodowym w danym momencie.

2.2. Procedura zbierania danych

2.2.1. Potencjalne źródła danych

Najłatwiej dostępnymi źródłami danych są wszelkie portale aukcyjne, które zawierają oferty sprzedaży samochodów. Zdecydowana większość ogłoszeń zawiera wszystkie potrzebne informacje o pojeździe, które zostały zdefiniowane w Sekcja 2.1. W szczególności ceny ustalane przez ogłoszeniodawców są zgodne z realiami panującymi na rynku, ponieważ to oni mają na niego największy wpływ. Do najpopularniejszych portali w Polsce należą między innymi:

- Allegro.pl
- Otomoto.pl
- Gratka.pl
- Autoplac.pl

Na potrzeby tego projektu zbiór danych zostanie stworzony na podstawie ogłoszeń zamieszczonych na portalu Otomoto.pl. Do pozyskania potrzebnych informacji zostanie wykorzystany autorski program (web scraper), który automatycznie pobierze i zapisze wszystkie wymagane pola z dostępnych na portalu ofert sprzedażowych.

2.2.2. Liczba potrzebnych przykładów

Na portalu Otomoto.pl w każdej chwili znajduje się około 200 tys. ofert sprzedaży z samochodami osobowymi. Można założyć, że jest to liczba co najmniej wystarczająca do wytrenowania systemu przewidującego ceny aut. Inną kwestią jest odpowiednio liczna reprezentacja każdej z dostępnych marek i modeli pojazdów. Przewiduje się, że w zbiorze treningowym powinno występować co najmniej kilka/kilkanaście przykładów dla każdego modelu auta, aby system był w stanie zwracać dla niego poprawne predykcje. W szczególności dotyczy to modeli rzadkich i unikatowych, których cena jest znacząco wyższa niż dla standardowych pojazdów.

2.2.3. Potencjalne problemy związane z gromadzeniem danych

Podczas gromadzenia zbioru danych mogą pojawić się potencjalne problemy związane z automatycznym pobieraniem informacji ze strony internetowej portalu Otomoto.pl. Nowoczesne strony internetowe wieloma sposobami blokują do siebie dostęp dla programów typu web scraper, ponieważ obsługa ich zapytań znacząco zwiększa koszty utrzymania witryny. Po wychwyceniu i obejściu potencjalnych pułapek, trzeba także będzie w sposób optymalny dobrać częstotliwość wysyłania zapytań do serwisu tak, aby program nie został przez niego zdemaskowany i zablokowany.

Drugim zagrożeniem są potencjalne problemy związane z brakiem istotnych informacji w ogłoszeniach. Mogą zdarzyć się oferty od ogłoszeniodawców, którzy na portalu wypełnili jedynie pola wymagane, a resztę informacji o pojeździe podają dopiero po kontakcie od potencjalnego klienta.

2.3. Zbieranie danych

2.3.1. Web scraper

Do implementacji programu wykorzystano język Python oraz następujące biblioteki:

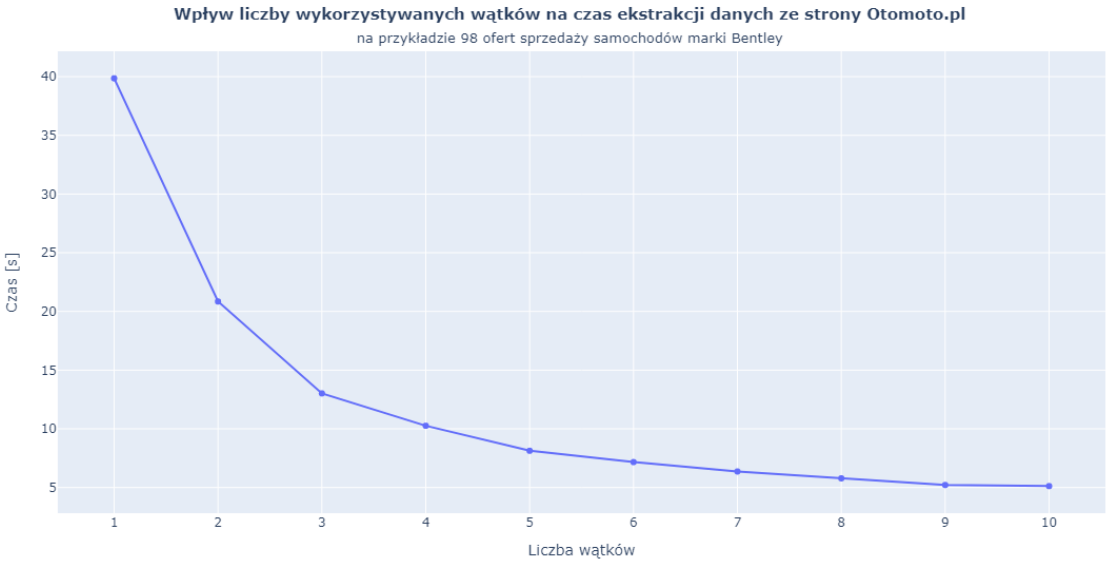
- HTTPX - obsługa protokołu HTTP (wysyłanie żądań i odbieranie odpowiedzi od serwisu)
- Selectolax - parsowanie kodu HTML oraz wyciąganie potrzebnych informacji za pomocą selektorów CSS
- Playwright - obsługa bezgłowej przeglądarki (Chromium), aby umożliwić przetwarzanie kodu JavaScript będącego częścią strony internetowej portalu.

Program automatycznie wyszukuje adresy URL do wszystkich dostępnych na portalu ogłoszeń, wydobywa z nich potrzebne informacje i na końcu zapisuje zebrane przykłady w formie tabelarycznej na dysku urządzenia. Sposób jego działania można opisać w następujących krokach:

1. Wydobycie nazw wszystkich dostępnych marek samochodów ze strony głównej portalu za pomocą przeglądarki Chromium oraz przetworzenie ich na właściwe adresy URL wyszukiwań.
2. Wydobycie nazw wszystkich dostępnych modeli dla marek z liczbą ofert większą niż 16000 (powód został opisany w następnym podrozdziale) oraz przetworzenie ich na właściwe adresy URL wyszukiwań.
3. Przejście przez wszystkie dostępne strony z ofertami dla każdego znalezionego wyszukiwania (konkretna marka pojazdu lub marka + model) w celu wydobycia adresu URL do każdej dostępnej tam oferty.
4. Wydobycie wszystkich potrzebnych informacji o pojeździe z każdego znalezionego adresu URL oferty.
5. Przetworzenie zgromadzonych informacji do postaci ramki danych i zapisanie jej do pliku CSV na dysku.

Do implementacji kroków 3. oraz 4. wykorzystano wielowątkowość, ponieważ są to zadania, których głównym ograniczeniem jest czas wymagany na operacje I/O. Takie podejście umożliwia pobieranie wielu stron w tym samym czasie wraz z równoległym przetwarzaniem i ekstrakcją danych z jednej, załadowanej już strony. Rysunek 2.1 przedstawia oszczędności czasowe, wynikające z wykorzystania wielowątko-

wości w implementacji kroku 3., na przykładzie scrapowania 98 ofert sprzedaży samochodów marki Bentley. Oczywiście na przedstawione statystyki mogły też mieć wpływ czynniki zewnętrzne takie jak obciążenie serwera w danym momencie. Widać jednak znaczne przyspieszenie wykonywania operacji, które spowalnia dopiero, gdy liczba wątków zbliża się do 10, na co wpływ mogło mieć osiągnięcie maksymalnej przepustowości łącza internetowego na urządzeniu klienta.



Rysunek 2.1: Wpływ liczby wykorzystywanych wątków na czas ekstrakcji danych ze strony Otomoto.pl

2.3.2. Napotkane przeszkody

W trakcie implementacji i uruchamiania web scrapera pojawiły się następujące problemy:

- Nazwy klas oraz ID większości elementów HTML na stronie portalu są losowe i często zmieniane. Są to atrybuty najczęściej wykorzystywane do odnajdywania interesujących tagów w celu wydobycia informacji ze strony. W tym przypadku trzeba było wykorzystać do tego celu selektory CSS.

- Maksymalny numer strony podczas przeglądania ofert, który jest dostępna dla użytkownika to 500. Podczas próby przejścia na stronę z wyższym numerem witryna przestaje wyświetlać oferty, przechodząc w tryb "wiecznego ładowania". Ta przeszkoda jest powodem, dla którego najpierw wydobywane są nazwy wszystkich marek (czasem także modeli) - ma to na celu zawęzić liczbę ofert dostępnych dla danego wyszukiwania, tak aby ich łączna liczba nie przekroczyła 16000 (32

* 500, gdzie 32 to liczba ofert na pojedynczej stronie). W ten sposób możemy uniknąć potrzeby przejścia na stronę z numerem większym niż ustalony próg.

- Adresy URL wyszukiwań dla niektórych marek oraz modeli są niezgodne z regułą, czyli są różne od faktycznej nazwy marki lub modelu. Aby zapobiec pominięciu tych pojazdów, ręcznie przygotowano listę takich przypadków i zaimplementowano ich oddzielną obsługę w kodzie programu.

- Średnio co kilkadziesiąt zapytań portal, pomimo poprawnego statusu odpowiedzi, wysyła stronę, na której niedostępna jest główna część, czyli lista ofert dla danego wyszukiwania lub szczegóły i opis w przypadku strony z konkretną ofertą. W takich sytuacjach zaimplementowano ponowne wysłanie zapytania do portalu.

- W przypadku zbyt dużej częstotliwości wysyłania zapytań do portalu po pewnym czasie adres IP nadawcy jest tymczasowo blokowany. To ograniczenie znacząco wydłużyło czas potrzebny na pobranie wszystkich dostępnych ofert. Próg detekcji i blokady programu przez portal starano się zwiększyć następującymi sposobami:

  – ustawienie losowego czasu pomiędzy wysyłanymi zapytaniami

  – ustawienia nagłówka wysyłanego zapytania na nagłówek wykorzystywany przez standardową przeglądarkę internetową.

  Finalna częstotliwość wysyłania zapytań została dobrana metodą prób i błędów - ustalono maksymalną liczbę wykorzystywanych wątków oraz przedział czasowy, spośród którego losowana jest długość przerwy przed wysłaniem kolejnego zapytania przez dany wątek.

2.3.3. Zebrany zbiór danych

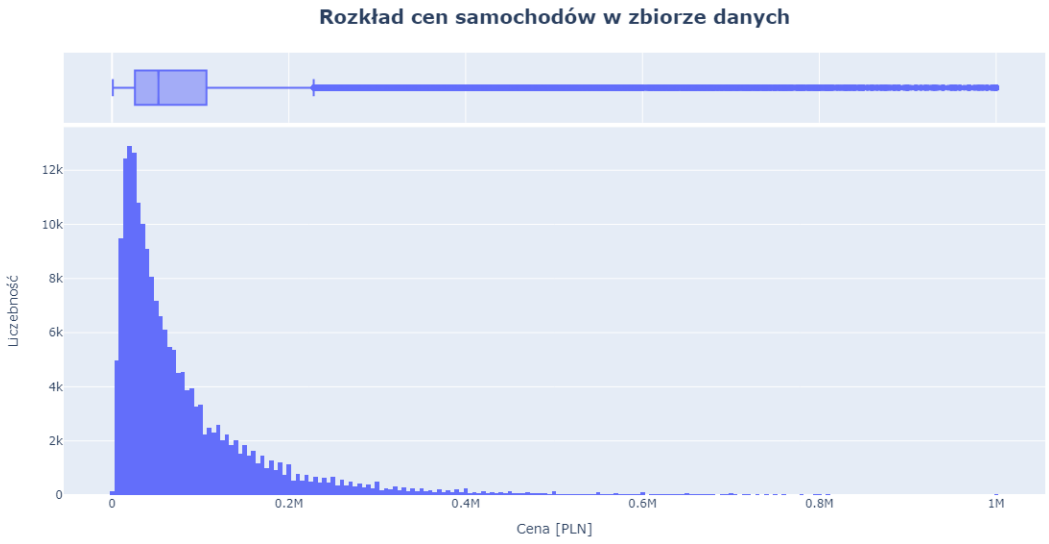
Łącznie ze strony Otomoto.pl zebrano informacje z 208205 ofert sprzedaży różnych samochodów. Finalna ramka danych posiada ponad 200 kolumn, gdzie zdecydowana większość z nich to kolumny binarne kodujące fakt występowania różnych opcji wyposażenia dodatkowego w pojeździe. Drugą grupę stanowią kolumny z informacjami o samym ogłoszeniu i sprzedawcy (m.in. ID, tytuł, data utworzenia oraz nazwa i typ sprzedawcy). Pozostałe kolumny zawierają wszystkie podstawowe informacje o pojeździe - część z nich została przedstawiona na rys. 2.2

marka_pojazdu	model_pojazdu	generacja	rok_produkcji	przebieg	rodzaj_paliwa	pojemnosc_skokowa	moc	skrzynia_biegow
Volvo	V70	III (2007-)	2010	304 000 km	Diesel	1 560 cm3	109 KM	Manualna
Honda	Accord	VII (2002-2008)	2005	236 000 km	Benzyna	1 998 cm3	155 KM	Manualna
Mercedes-Benz	Klasa X	NaN	2019	73 000 km	Diesel	2 987 cm3	258 KM	Automatyczna
Toyota	Avensis	II (2003-2009)	2005	220 000 km	Benzyna	1 794 cm3	129 KM	Manualna
Ford	C-MAX	II (2010-)	2012	179 058 km	Diesel	1 997 cm3	140 KM	Manualna
Peugeot	208	II (2019-)	2023	1 km	Benzyna	1 199 cm3	75 KM	Manualna
Kia	Sportage	III (2010-2015)	2018	100 420 km	Benzyna	1 591 cm3	132 KM	Manualna
Toyota	Auris	I (2006-2012)	2008	198 500 km	Benzyna+LPG	1 398 cm3	97 KM	Manualna
Citroën	C5	III (2008-)	2010	171 500 km	Benzyna	1 598 cm3	156 KM	Manualna
Volvo	V60	I (2011-2018)	2011	279 000 km	Diesel	1 984 cm3	163 KM	Manualna
Honda	CR-V	IV (2012-2018)	2013	103 000 km	Benzyna	1 997 cm3	155 KM	Manualna
Opel	Astra	K (2015-2021)	2020	12 000 km	Benzyna	1 199 cm3	130 KM	Manualna
Peugeot	208	I (2012-2019)	2016	115 055 km	Benzyna	1 199 cm3	82 KM	Manualna
Fiat	Tipo	II (2016-)	2021	16 600 km	Benzyna	999 cm3	100 KM	Manualna
Audi	A6	C7 (2011-2018)	2016	176 000 km	Diesel	2 967 cm3	218 KM	Automatyczna
BMW	X3	F25 (2010-)	2011	298 765 km	Diesel	1 995 cm3	184 KM	Manualna
Ford	Mondeo	Mk5 (2014-)	2018	28 000 km	Benzyna	1 499 cm3	160 KM	Automatyczna
Toyota	RAV4	IV (2012-2018)	2016	127 000 km	Diesel	1 995 cm3	143 KM	Manualna
Seat	Ibiza	V (2017-)	2020	99 990 km	Benzyna	999 cm3	80 KM	Manualna
Toyota	Camry	NaN	1992	220 000 km	Benzyna	2 164 cm3	136 KM	Manualna

Rysunek 2.2: Podzbiór podstawowych kolumn opisujących pojazdy dla 20 przykładowych ofert pobranych ze strony Otomoto.pl.

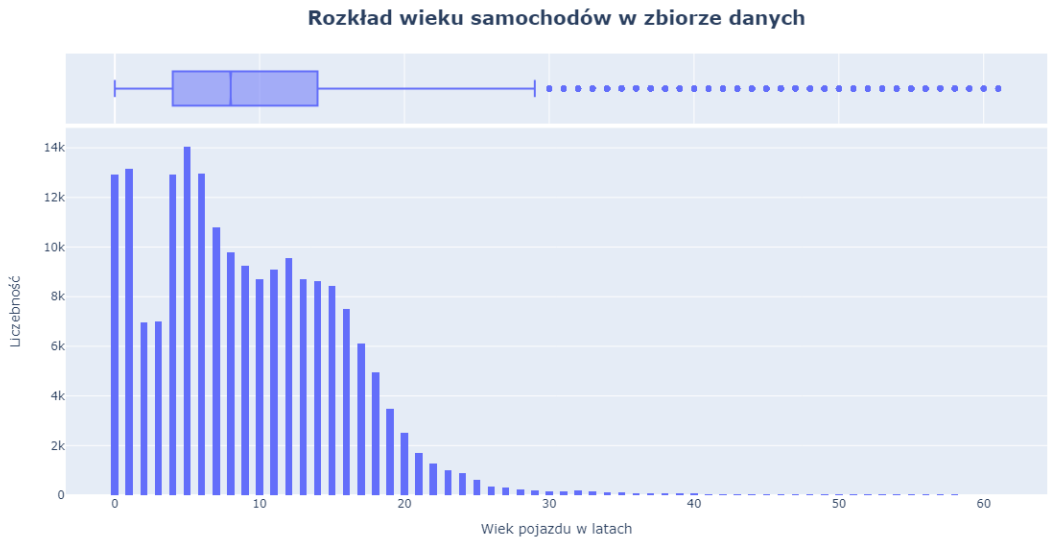
2.4. Wizualizacja i analiza zbioru danych

Ceny samochodów w praktyce nie mają górnej granicy, dlatego ich rozkład charakteryzuje się tzw. długim ogonem. Histogram cen aut ze zbioru danych został przedstawiony na rys. 2.3. Niewielka liczba bardzo drogich samochodów w zbiorze danych może powodować zwiększone błędy predykcji dla takich przykładów.



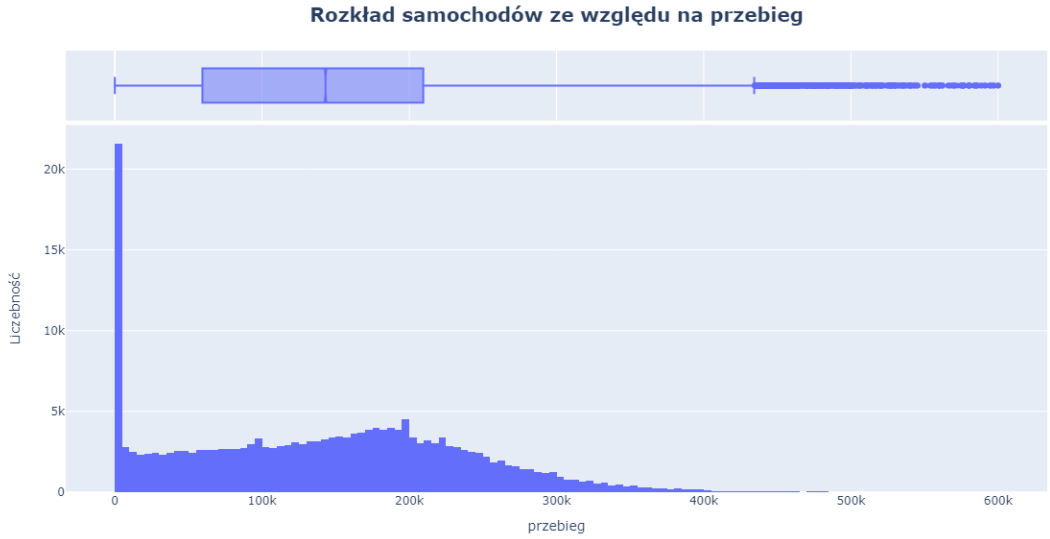
Rysunek 2.3: Rozkład cen samochodów ze zbioru danych.

Rysunek 2.4 przedstawia rozkład wieku samochodów w zbiorze danych. Ciekawym zjawiskiem jest spadek liczby ofert dla samochodów dwuletnich oraz trzyletnich, co może być spowodowane tym, że na tym etapie ich wartość traci najbardziej. Rozkład ten także posiada długi ogon, co również stwarza zagrożenie pogorszenia jakości predykcji dla samochodów bardzo starych z niewielką liczbą przykładów w zbiorze.



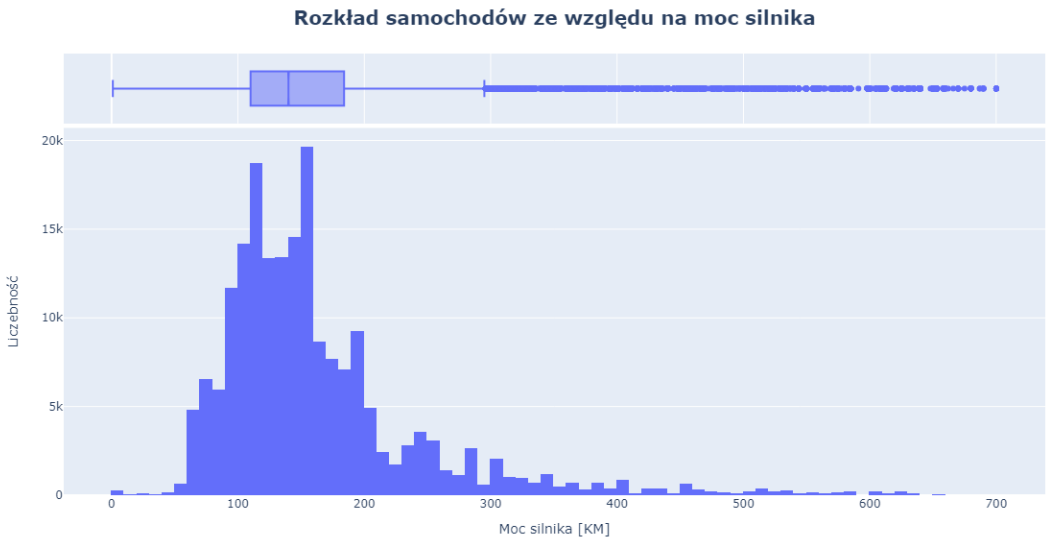
Rysunek 2.4: Rozkład wieku samochodów w zbiorze danych.

Rysunek 2.5 przedstawia rozkład przebiegu samochodów w zbiorze danych. Widać na nim zwiększoną liczbę ofert dla samochodów z przebiegiem zbliżającym się 100 tysięcy kilometrów oraz do 200 tysięcy kilometrów. Histogram ten również charakteryzuje się długim ogonem.



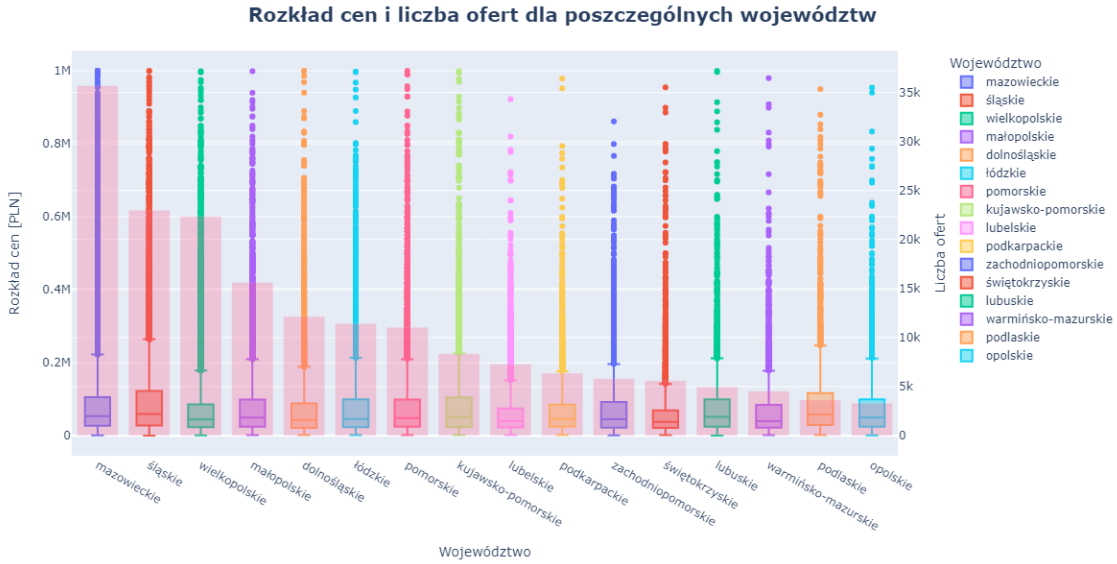
Rysunek 2.5: Rozkład samochodów ze względu na przebieg.

Moc silnika jest kolejną cechą samochodów, dla której ciężko ustalić górną granicę wartości. Widać to na rys. 2.6 w postaci długiego ogona rozkładu tej cechy. Można także zauważyć, że zdecydowana większość pojazdów ze zbioru danych ma moc silnika w przedziale od 100 do 200 KM.



Rysunek 2.6: Rozkład samochodów ze względu na moc silnika.

Rysunek 2.7 przedstawia rozkład cen samochodów oraz liczbę ofert z podziałem na poszczególne województwa. Nie widać znaczących różnic w cenach pojazdów pomiędzy regionami. Z kolei w przypadku liczby ofert zdecydowaną przewagę ma województwo mazowieckie, z którego pochodzi ponad 35 tysięcy ogłoszeń.

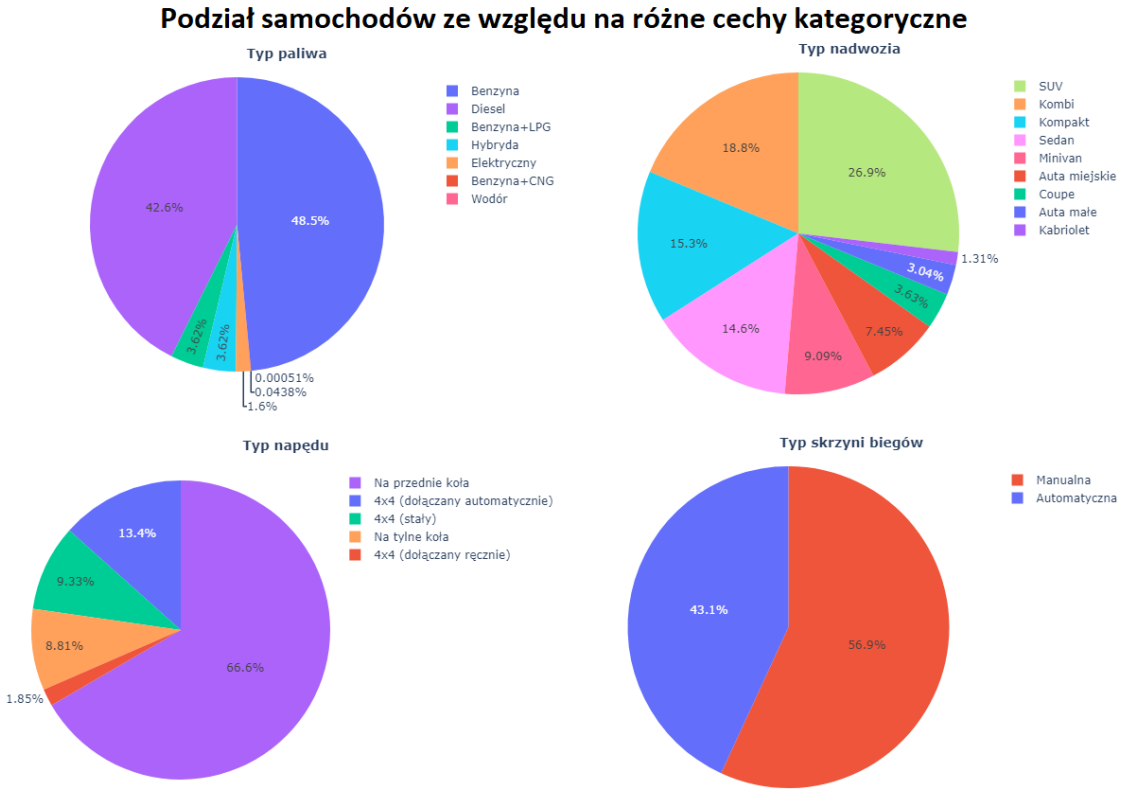


Rysunek 2.7: Rozkład cen i liczba ofert dla poszczególnych województw.

Rysunek 2.8 przedstawia podział samochodów ze względu na 4 cechy kategoryczne: rodzaj paliwa, rodzaj nadwozia, typ napędu oraz typ skrzyni biegów. Na podstawie zamieszczonych tam wykresów kołowych można wysnuć następujące wnioski:

- Auta spalinowe nadal stanowią ponad 90% rynku samochodowego

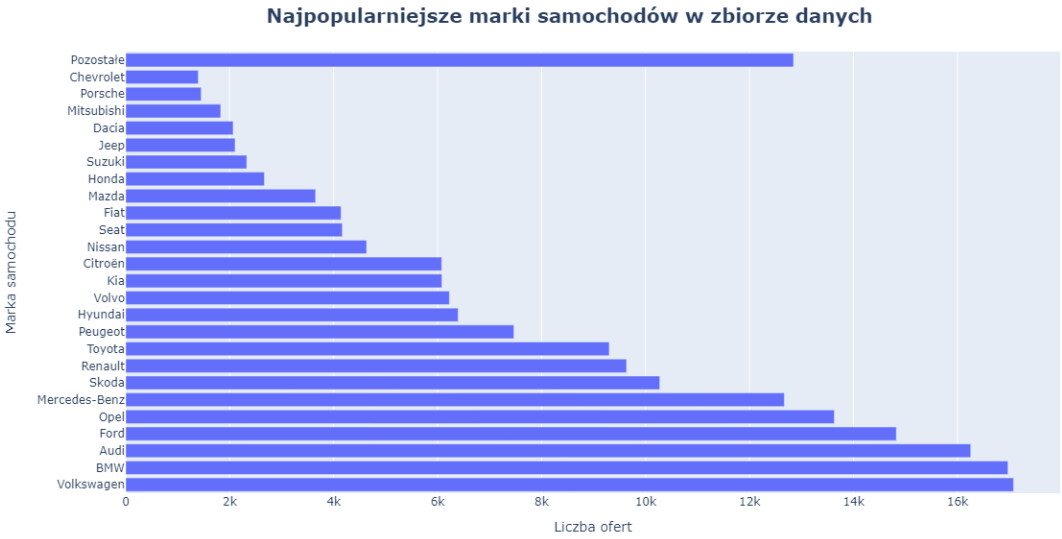
- Najbardziej popularna w Polsce jest manualna skrzynia biegów (57%)
- Podobnie jak typ napędu na przednie koła, który jest ponad dwukrotnie częściej spotykany niż drugi w kolejności napęd 4x4
- Najbardziej liczną grupą aut są, wciąż zyskujące na popularności, samochody typu SUV.



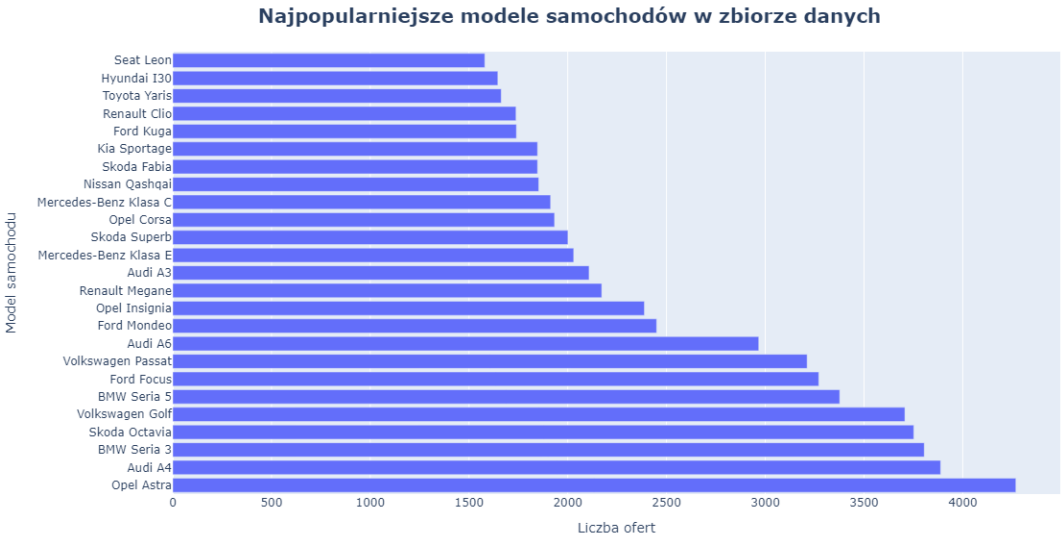
Rysunek 2.8: Podział samochodów ze względu na różne cechy kategoryczne.

Z rys. 2.9 można odczytać nazwy 25 najpopularniejszych marek samochodów na portalu Otomoto.pl. W czołówce zestawienia znajdują się następujący producenci: Volkswagen, BMW, Audi, Ford oraz Opel.

Z kolei na rys. 2.10 znajdują się nazwy najpopularniejszych modeli samochodów. Top 3 zestawienia stanowią: Opel Astra, Audi A4 oraz BMW Serii 3.



Rysunek 2.9: Najpopularniejsze marki samochodów w zbiorze danych.

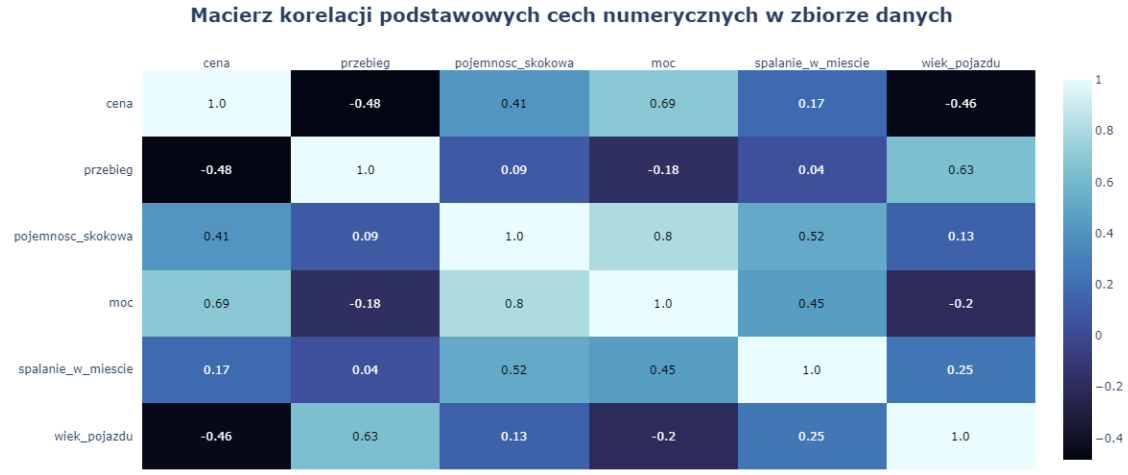


Rysunek 2.10: Najpopularniejsze modele samochodów w zbiorze danych.

Rysunek 2.11 przedstawia liczbę ofert, w których pojawiły się najpopularniejsze wyposażenia dodatkowe samochodów. Do najczęściej spotykanych zaliczają się: system ABS, poduszki powietrzne dla pasażera oraz radio.

Rysunek 2.11: Najpopularniejsze wyposażenie dodatkowe w zbiorze danych.

Obliczono także wartości korelacji pomiędzy podstawowymi cechami numerycznymi ze zbioru danych. Wyniki obliczeń zostały przedstawione na rys. 2.12 w postaci macierzy korelacji. Największy dodatni związek z ceną pojazdu ma moc silnika (wartość równa 0.69). Z kolei statystycznie najbardziej ujemny wpływ na wartość samochodu mają duże wartości przebiegu oraz wieku pojazdu (wartości korelacji równe odpowiednio -0.48 oraz -0.46).



Rysunek 2.12: Macierz korelacji podstawowych cech numerycznych w zbiorze danych.

Sprawdzono także wpływ wyposażenia dodatkowego na ceny samochodów. Wykorzystano do tego celu współczynnik korelacji punktowo-dwuseryjnej (ang. Point Biserial Correlation). Wyniki obliczeń zostały przestawione na rys. 2.13 w formie tabelarycznej. Znajduje się tam 25 typów wyposażenia dodatkowego, dla których wartość obliczonej korelacji była największa. W trakcie obliczeń odrzucono te wyniki, dla których

stopień istotności statystycznej był poniżej założonego progu (P value było większe od 0.05).

	Nazwa wyposażenia dodatkowego	Wartość korelacji z ceną pojazdów
0	kamera_parkowania_tyl	0.333753
1	lampy_przednie_w_technologii_led	0.329514
2	lane_assist_kontrola_zmiany_pasa_ruchu	0.319701
3	elektrycznie_ustawiany_fotel_pasazera	0.317009
4	elektrycznie_ustawiany_fotel_kierowcy	0.299964
5	apple_carplay	0.299734
6	asystent_swiatel_drogowych	0.295564
7	asystent_czujnik_martwego_pola	0.292518
8	siedzenie_z_pamiecia_ustawienia	0.285823
9	android_auto	0.274344
10	ladowanie_bezprzewodowe_urzadzen	0.273655
11	felgi_aluminiowe_od_21	0.271672
12	tapicerka_skorzana	0.270078
13	kamera_panoramiczna_360	0.267623
14	zmiana_biegow_w_kierownicy	0.262825
15	park_assistant_asystent_parkowania	0.262796
16	system_ostrzegajacy_o_mozliwej_kolizji	0.262585
17	system_nawigacji_satelitarnej	0.262036
18	system_rozpoznawania_znakow_drogowych	0.253638
19	felgi_aluminiowe_20	0.247765
20	system_powiadamiania_o_wypadku	0.245901
21	lampy_tylne_w_technologii_led	0.243361
22	kontrola_odleglosci_od_poprzedzajacego_pojazdu	0.242901
23	dostep_do_internetu	0.238678
24	zawieszenie_regulowane	0.234741

Rysunek 2.13: Cechy binarne wyposażenia dodatkowego najbardziej skorelowane z ceną pojazdów.

Obliczono także współczynniki korelacji punktowo-dwuseryjnej pomiędzy pozostałymi cechami binarnymi a cenami pojazdów. Wyniki tego eksperymentu przedstawiono rys. 2.14. W tym przypadku także odrzucono wyniki, które okazały się nieistotne statystycznie.

	Nazwa cechy	Wartość korelacji z ceną pojazdów
0	faktura_vat	0.415107
1	leasing	0.297936
2	bezwypadkowy	0.191757
3	pierwszy_wlasciciel_od_nowosci	0.136075
4	ma_numer_rejestracyjny	-0.072085
5	serwisowany_w_aso	0.040861
6	zarejestrowany_w_polsce	-0.030927
7	filtr_czastek_stalych	0.018143
8	kierownica_po_prawej_anglik	-0.017907
9	tuning	0.014688

Rysunek 2.14: Cechy binarne wyposażenia dodatkowego najbardziej skorelowane z ceną pojazdów.

2.5. Wstępne przetwarzanie danych

Etap wstępnego przetwarzania danych składał się z następujących części:

1. Czyszczenie danych:
    - Usunięcie zduplikowanych wierszy ze zbioru danych.
    - Odrzucenie kolumn z informacjami nieprzydatnymi podczas predykcji (np. ID oferty, adres URL, czy numer VIN pojazdu) oraz kolumn, gdzie wartości brakujące stanowiły prawie 100% całości (m.in. emisja CO^2 oraz czas ładowania baterii).
    - Usunięcie ofert z samochodami uszkodzonymi, ponieważ ich ceny znacznie odbiegają od normy, a w informacjach pobranych ze strony ciężko znaleźć cechy umożliwiające oszacowanie wartości szkody.
    - Zamiana wszystkich cen na tę samą walutę (PLN) zgodnie z kursami obowiązującymi w dniu wystawienia oferty.
    - Usunięcie jednostek i zamiana typów kolumn do wartości numerycznych.
    - Przetworzenie kolumn dotyczących wyposażenia dodatkowego do postaci binarnej i wypełnienie brakujących wartości zerami.
    - Przetworzenie pozostałych kolumn typu prawda/fałsz do postaci binarnej i wypełnienie brakujących wartości.

2. Ekstrakcja cech:

    - Dodanie kolumny wskazującej na brak uwzględnienia informacji o wyposażeniu dodatkowym w ogłoszeniu.
    - Obliczenie wieku samochodu na podstawie roku produkcji i daty wystawienia oferty.
    - Obliczenie pozostałej liczby miesięcy obowiązywania gwarancji producenta.
    - Ekstrakcja nazwy województwa z tekstu zawierającego adres oferty. Dla części ofert wykorzystano do tego celu zewnętrzne API, ponieważ w adresie oferty nie zawarto informacji o województwie.
    - Wydobycie dwóch cech numerycznych (liczba generacji danego modelu oraz historyczna kolejność generacji) z kolumny kategorycznej zawierającej informacje o generacji modelu samochodu. Pozwoli to uniknąć kodowania tej cechy, co znacznie zwiększyłoby wymiarowość zbioru danych.

3. Uzupełnienie brakujących wartości:

    - Za pomocą mediany w kolumnach: przebieg, liczba drzwi, liczba miejsc, pojemność silnika, moc silnika oraz spalanie.
    - Za pomocą najczęściej występujących wartości w kolumnach: rodzaj skrzyni biegów, rodzaj napędu, rodzaj koloru karoserii.
    - Za pomocą wartości “Nieznany” dla cechy oznaczającej kraj pochodzenia pojazdu.
    - Zrezygnowano z bardziej zaawansowanych metod wypełniania braków (np. KNNImputer) ze względu na problemy z pamięcią.

4. Kodowanie cech kategorycznych:

    - Wykorzystano do tego celu metodę one-hot-encoding, gdzie wszystkim najmniej licznym kategoriom, których liczba wystąpień była poniżej określonego progu, została przypisana jedna wspólna wartość. Zakodowano następujące cechy: marka, model, typ nadwozia, rodzaj paliwa, rodzaj skrzyni biegów, typ sprzedawcy, kolor, rodzaj koloru, rodzaj napędu, kraj pochodzenia oraz województwo.

5. Standaryzacja cech numerycznych:

    - Wykorzystano do tego celu standardową metodę, polegającą na odjęciu wartości średniej i podzieleniu przez odchylenie standardowe. Operację tę

zastosowano do następujących kolumn: przebieg, moc silnika, spalanie, pojemność silnika, wiek pojazdu, liczba drzwi, liczba miejsc, długość gwarancji dealerskiej oraz długość gwarancji producenta.

Po etapie wstępnego przetwarzania wynikowa ramka danych zawiera 1168 kolumn.

2.6. Podział zbioru danych

Zbiór danych został podzielony na następujące podzbiory: treningowy, walidacyjny oraz testowy. Udział każdego ze zbiorów oraz dokładne liczby przykładów w każdym zbiorze zostały przedstawione na rys. 2.15.

	Liczba próbek	Procent udziału
Zbiór treningowy	137298	70.00%
Zbiór walidacyjny	29421	15.00%
Zbiór testowy	29421	15.00%

Rysunek 2.15: Podstawowe statystyki dla zbiorów: treningowego, walidacyjnego oraz testowego.

Dodatkowo podziału zbioru danych dokonano w sposób warstwowy, zachowując zbliżony udział poszczególnych modeli samochodów w każdym z podzbiorów, co zostało przedstawione na rys. 2.16.

	Nazwa modelu	Zbiór treningowy	Zbiór walidacyjny	Zbiór testowy
0	Opel_Astra	2.176%	2.175%	2.179%
1	Audi_A4	1.983%	1.982%	1.982%
2	BMW_Seria_3	1.940%	1.941%	1.941%
3	Skoda_Octavia	1.913%	1.914%	1.914%
4	Volkswagen_Golf	1.890%	1.890%	1.890%
5	BMW_Seria_5	1.722%	1.720%	1.723%
6	Ford_Focus	1.668%	1.669%	1.665%
7	Volkswagen_Passat	1.637%	1.638%	1.638%
8	Audi_A6	1.513%	1.513%	1.513%
9	Ford_Mondeo	1.249%	1.247%	1.251%
10	Opel_Insignia	1.218%	1.217%	1.217%
11	Renault_Megane	1.107%	1.108%	1.108%
12	Audi_A3	1.075%	1.074%	1.074%
13	Mercedes-Benz_Klasa_E	1.035%	1.033%	1.037%
14	Skoda_Superb	1.020%	1.020%	1.020%
15	Opel_Corsa	0.985%	0.986%	0.986%
16	Mercedes-Benz_Klasa_C	0.975%	0.975%	0.975%
17	Nissan_Qashqai	0.945%	0.945%	0.945%
18	Skoda_Fabia	0.942%	0.942%	0.942%
19	Kia_Sportage	0.942%	0.942%	0.942%

Rysunek 2.16: Porównanie udziału różnych modeli samochodów w zbiorach: treningowym, walidacyjnym oraz testowym.

Stworzono 3 następujące zestawy podzbiorów:

- SPLIT 1 - podział oryginalny (opisany powyżej) bez standaryzacji cech numerycznych podczas wstępnego przetwarzania.
- SPLIT 2 - podział identyczny do oryginalnego, ale z z zastosowaniem standaryzacji cech numerycznych podczas wstępnego przetwarzania.
- SPLIT 3 - podział z uwzględnieniem wszystkich operacji przetwarzania (włącznie ze standaryzacją), ale z połączeniem zbioru walidacyjnego ze zbiorem treningowym.

3. TRENOWANIE I TESTY MODELI

3.1. Wybór modelu i metryk ewaluacji

Zadaniem postawionym przed projektowanym systemem jest predykcja cen samochodów, czyli zmienna zależna należy do dziedziny liczb rzeczywistych większych od 0. Oznacza to, że rozwiązywany problem jest problemem regresji. Z kolei podstawowym sposobem treningu modeli regresyjnych jest uczenie nadzorowane i zostanie ono także zastosowane w ramach niniejszego projektu. Do tego celu zostanie wykorzystany zbiór danych, który został opisany w Rozdział 2.

Do podstawowych typów modeli regresyjnych należą takie algorytmy jak: regresja liniowa, K najbliższych sąsiadów, Support Vector Machines, czy drzewa decyzyjne. Jednak w tym przypadku do budowy modelu zostanie wykorzystany algorytm drzew decyzyjnych wzmacnianych gradientowo, którego efektywna implementacja jest dostępna w pakiecie XGBoost. Jest on uznawany za jeden z najbardziej solidnych algorytmów w kontekście pracy z danymi tabelarycznymi, czego dowodem może być jego częste stosowanie przez zwycięzców konkursów na platformach takich jak Kaggle. Do jego licznych zalet należą m.in. wysoka wydajność na dużych zbiorach danych, wysoka skuteczność predykcyjna oraz wewnętrzne mechanizmy regularyzacyjne, które zwiększają jego odporność na szum w danych i zapobiegają przeuczeniu się.

Algorytm XGBoost posiada wiele hiperparametrów, które znacząco wpływają na proces dopasowywania się modelu do danych treningowych oraz jego późniejszą skuteczność dokonywania predykcji. Do wstępnych testów modeli na każdym z przygotowanych zestawów danych wybrano następującą konfigurację:

- $max\_depth = 15$ - maksymalna głębokość pojedynczego estymatora
- $learning\_rate = 0.3$ - współczynnik uczenia
- $subsample = 0.9$ - procentowy udział wylosowanych próbek treningowych używanych do trenowania każdego estymatora
- $colsample\_bytree = 0.9$ - procentowy udział wylosowanych cech wejściowych używanych do trenowania każdego estymatora
- $gamma = 0.5$ - minimalna redukcja funkcji kosztu wymagana do utworzenia nowego liścia w drzewie.

Liczba estymatorów ($n\_estimators$) zostanie dobrana na podstawie krzywej uczenia się poszczególnych modeli. Pozostałe hiperparametry przyjmą wartości domyślne.

Jako funkcję kosztu w trakcie treningu modeli zostanie wykorzystana funkcja Squared Error (błąd kwadratowy). Inną potencjalną funkcją kosztu mógłby być błąd bezwzględny (Absolute Error), jednak zdecydowano się wybrać tę pierwszą, aby uniknąć dużych błędów predykcji np. dla pojazdów unikatowych. Dzięki takiemu podejściu błędy te będą miały większy wpływ na proces dopasowywania się modelu na danych treningowych, czyli podczas dodawania kolejnych estymatorów do zespołu.

Do oceny działania modeli na zbiorach walidacyjnych oraz testowych zostaną wykorzystane następujące metryki błędu: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), R^2 score oraz histogram błędów predykcji. Różnica wartości pomiędzy RMSE oraz MAE pozwoli określić, czy model często popełnia duże błędy w swoich estymacjach (przypadek gdy RMSE » MAE). Metryka MAPE wskaże, jaki jest średni błąd procentowy pomiędzy uzyskanymi predykcjami a cenami rzeczywistymi. Z kolei R^2 score pozwoli określić, o ile wytrenowany model jest lepszy od prostego estymatora, zwracające zawsze cenę średnią ze zbioru treningowego. Histogram błędów posłuży jako wizualna metryka oceny modelu - w sytuacji idealnej powinien on przypominać rozkład normalny ze średnią w okolicach zera. Powinien być on także symetryczny - w przeciwnym razie będzie można dostrzec, czy model częściej niedoszacowuje, czy przeszacowuje ceny pojazdów.

3.2. Trening modelu na zestawie danych SPLIT 1

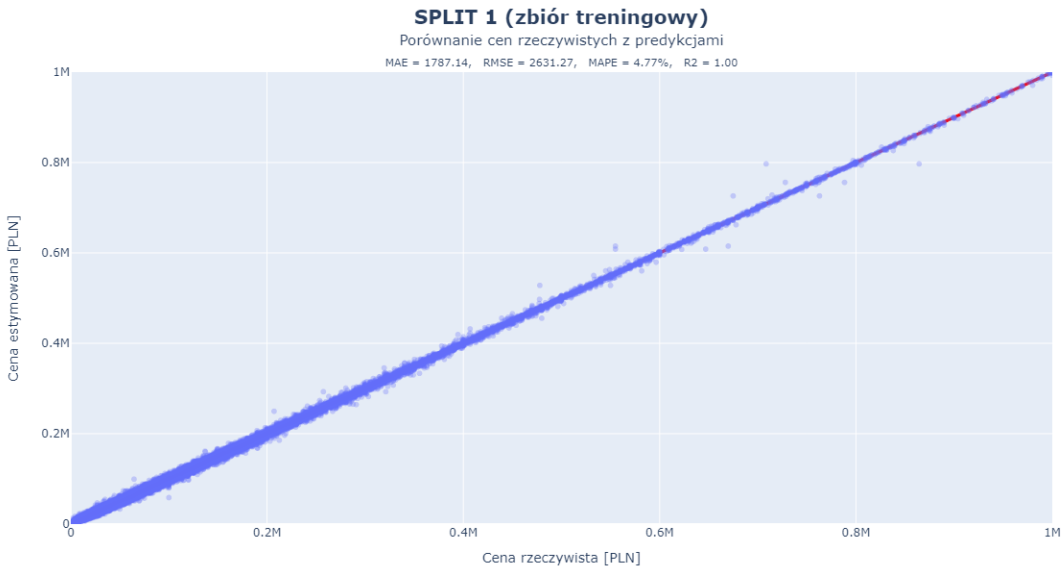
Rysunek 3.1 przedstawia krzywe uczenia się modelu dopasowywanego do zestawu danych SPLIT 1 (MODEL 1). Widzimy, że po dodaniu kilkunastu estymatorów do zespołu wartości funkcji kosztu na zbiorze walidacyjnym gwałtownie przestają maleć wraz z dalszą rozbudową modelu. W tym samym czasie spadek wartości funkcji kosztu na zbiorze treningowym również spowalnia, jednak dzieje się to zdecydowanie wolniej. Znaleziona optymalna liczba estymatorów wynosi 309 - dalsze iteracje nie przyniosły znaczących różnic w wartościach funkcji kosztu na zbiorze walidacyjnym.

Zaimplementowane w algorytmie XGBoost mechanizmy regularyzacyjne (wykorzystane m.in. za pomocą hiperparametrów $gamma$ oraz $subsample$) sprawiają, że model nie ulega przetrenowaniu pomimo dodawania zbędnych estymatorów w kolejnych iteracjach. Potencjalnie można by doprowadzić do przetrenowania modelu poprzez wyłączenie wspomnianych mechanizmów oraz zwiększenie maksymalnej głębokości drzew za pomocą hiperparametru max_depth.

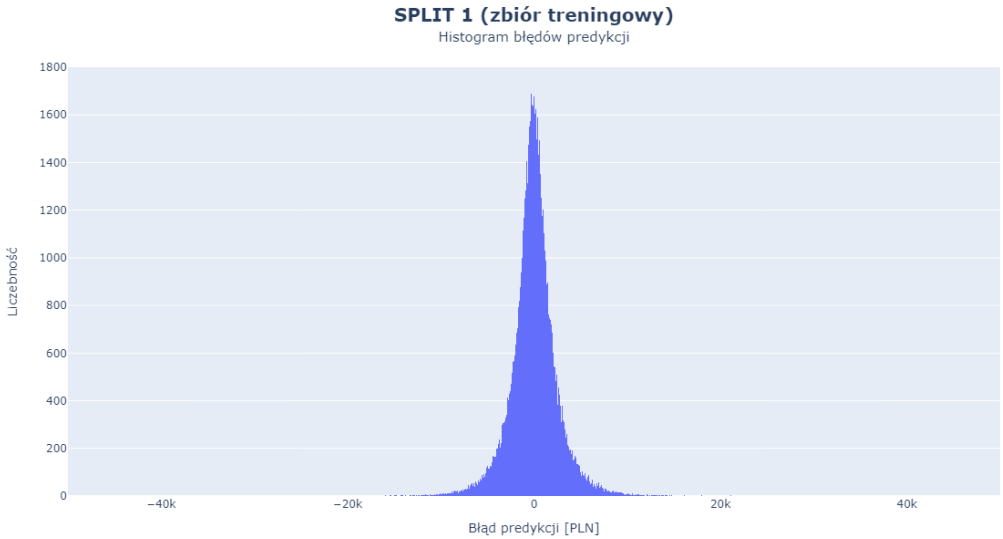
Finalnie dokonano ponownego dopasowania modelu, ustawiając hiperparametr $n\_estimators$ na wartość 309. Trening w takiej konfiguracji trwał 6 minut i 37 sekund.

Rysunek 3.1: Krzywa uczenia się na zestawie danych SPLIT 1.

Na rys. 3.2 oraz rys. 3.3 przedstawiono wyniki ewaluacji modelu na zbiorze treningowym. Estymacje modelu różnią się średnio o około 1800 zł od cen rzeczywistych, co stanowi jedynie niecałe 5% wartości pojazdów. Współczynnik R^2 jest bliski 1, która stanowi maksymalną możliwą wartość. Histogram błędów przypomina rozkład normalny i jest symetryczny względem wartości średniej, która leży w okolicach 0.

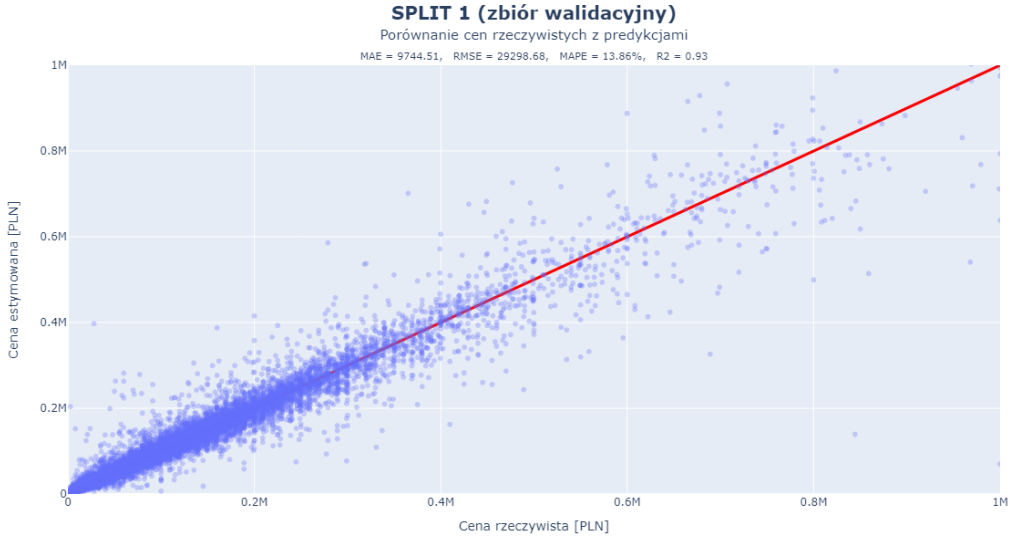


Rysunek 3.2: Porównanie cen rzeczywistych z predykcjami na próbkach treningowych z zestawu danych SPLIT 1.

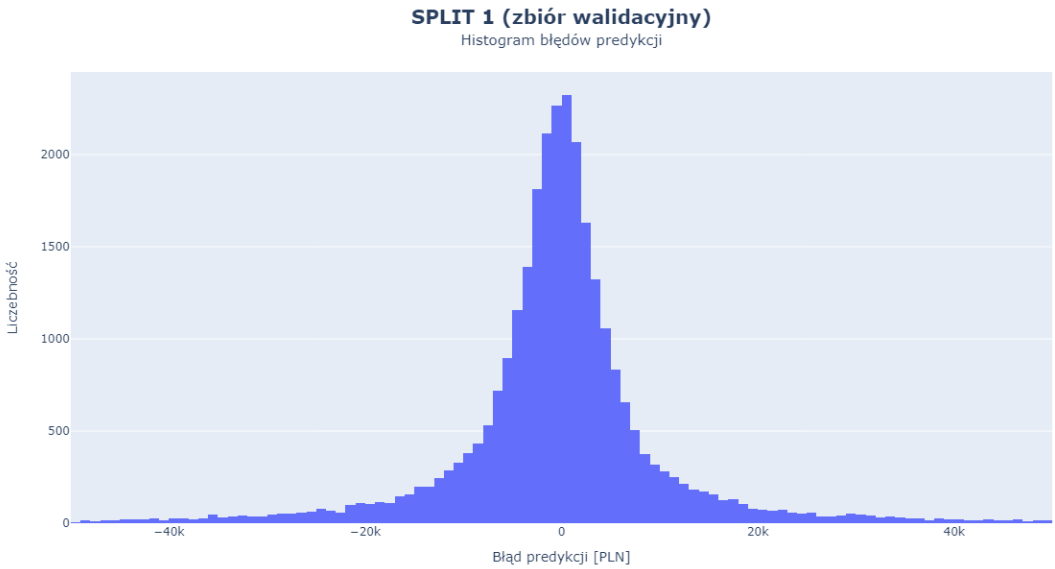


Rysunek 3.3: Histogram błędów predykcji na próbkach treningowych z zestawu danych SPLIT 1.

Na rys. 3.4 oraz rys. 3.5 przedstawiono wyniki ewaluacji modelu na zbiorze walidacyjnym. Skuteczność modelu jest znacznie gorsza niż na zbiorze treningowym - średnia różnica pomiędzy estymacjami a cenami rzeczywistymi urosła już do około 9745 zł, co stanowi prawie 14% wartości pojazdów. Jednak wartość współczynnika R^2 wynosi 0.93, co nadal stanowi wynik bardzo dobry. Histogram błędów w dalszym ciągu przypomina rozkład normalny i jest symetryczny względem 0. Wzrosło jedynie odchylenie standardowe błędów, przez co histogram jest bardziej rozciągnięty na boki.



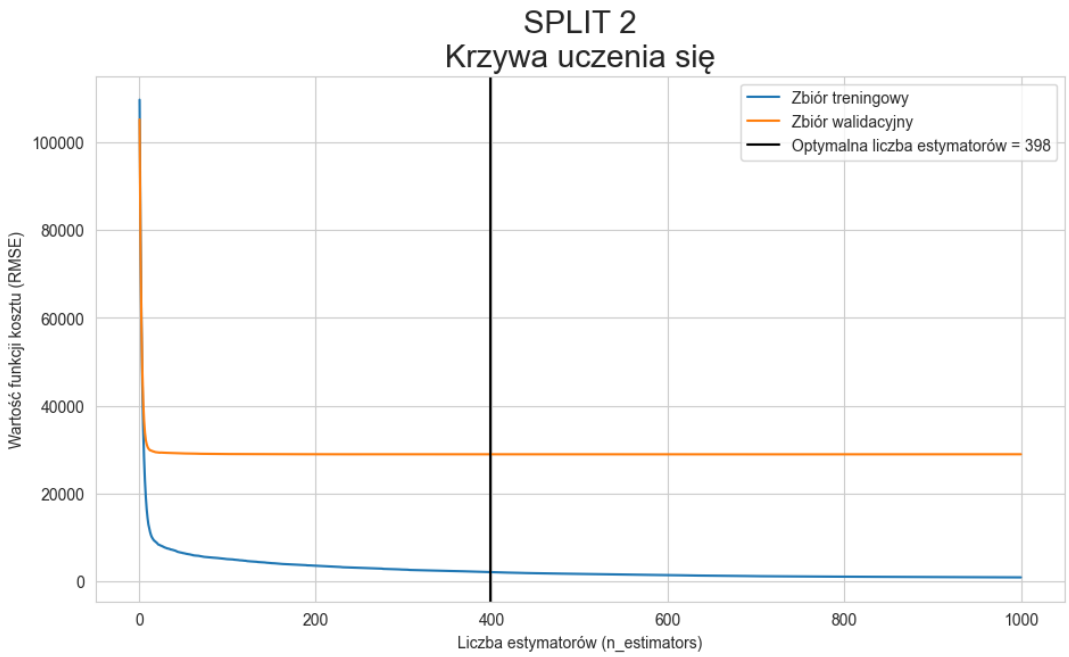
Rysunek 3.4: Porównanie cen rzeczywistych z predykcjami na próbkach walidacyjnych z zestawu danych SPLIT 1.



Rysunek 3.5: Histogram błędów predykcji na próbkach walidacyjncyh z zestawu danych SPLIT 1.

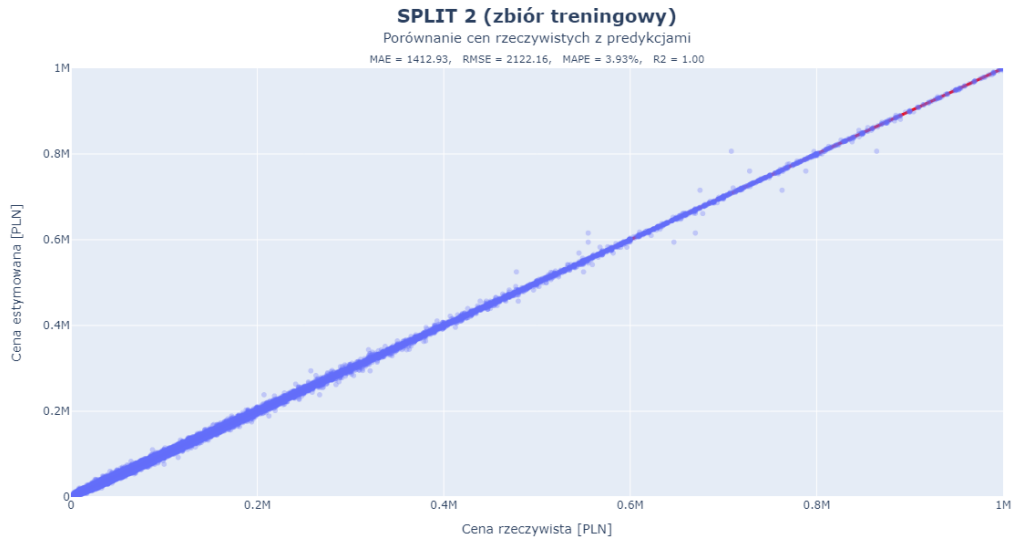
3.3. Trening modelu na zestawie danych SPLIT 2

Rysunek 3.6 przedstawia krzywe uczenia się modelu dopasowywanego do zestawu danych SPLIT 2 (MODEL 2). Sytuacja jest analogiczna jak w przypadku zestawu danych SPLIT 1 - wartości funkcji kosztu na zbiorze walidacyjnym gwałtownie przestają maleć już po kilku iteracjach dodawania estymatorów do zespołu. W tym samym czasie spadek wartości funkcji kosztu na zbiorze treningowym również spowalnia, jednak dzieje się to zdecydowanie wolniej. Znaleziona optymalna liczba estymatorów wynosi 398, jednak jest to wartość mocno orientacyjna, ponieważ już po kilkudziesięciu iteracjach wartość funkcji kosztu na zbiorze walidacyjnym pozostaje na stałym poziomie, zatem można by wybrać dowolną wartość ze sprawdzanego przedziału (miało by to jedynie wpływ na czas inferencji modelu). Trening modelu dla takiej konfiguracji trwał 8 minut i 33 sekundy. Czas dopasowywania modelu wzrósł o prawie 2 minuty względem zestawu SPLIT 1, co jest spowodowane większą liczbą estymatorów, które w algorytmie XGBoost są dodawane iteracyjnie.

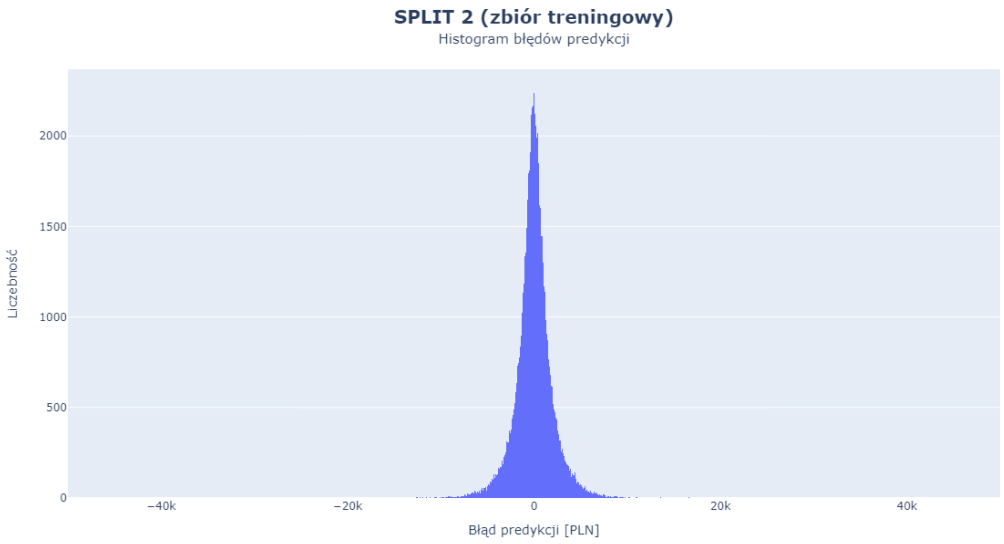


Rysunek 3.6: Krzywa uczenia się na zestawie danych SPLIT 2.

Na rys. 3.7 oraz rys. 3.8 przedstawiono wyniki ewaluacji modelu na zbiorze treningowym. Dokładność estymacji modelu poprawiła się względem wyników uzyskanych na zbiorze treningowym w zestawie SPLIT 1 - średni błąd bezwzględny zmalał do poziomu 1413 zł, a średni błąd procentowy do niecałych 4%. Histogram błędów w dalszym ciągu przyjmuje pożądaną postać, czyli przypomina rozkład normalny i jest symetryczny względem 0.

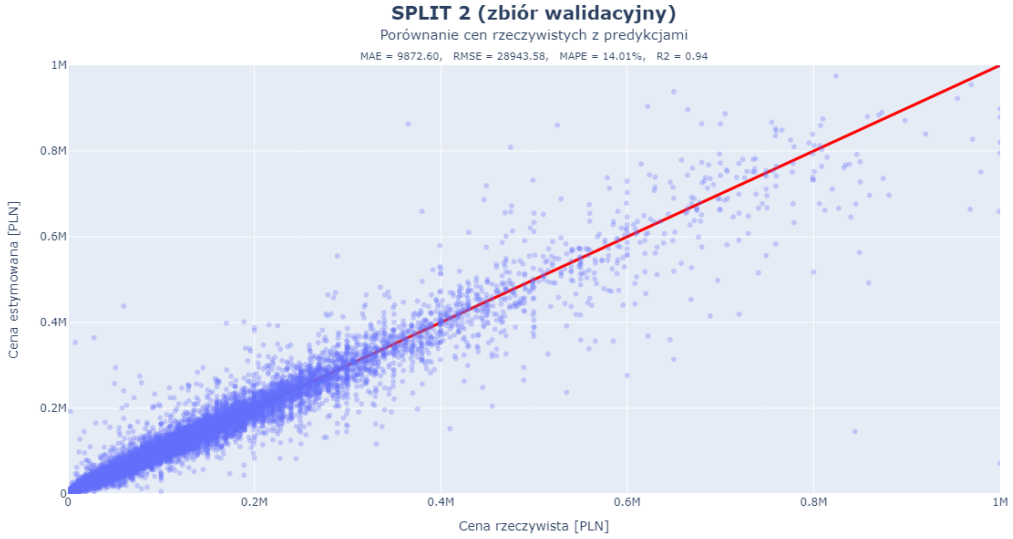


Rysunek 3.7: Porównanie cen rzeczywistych z predykcjami na próbkach treningowych z zestawu danych SPLIT 2.

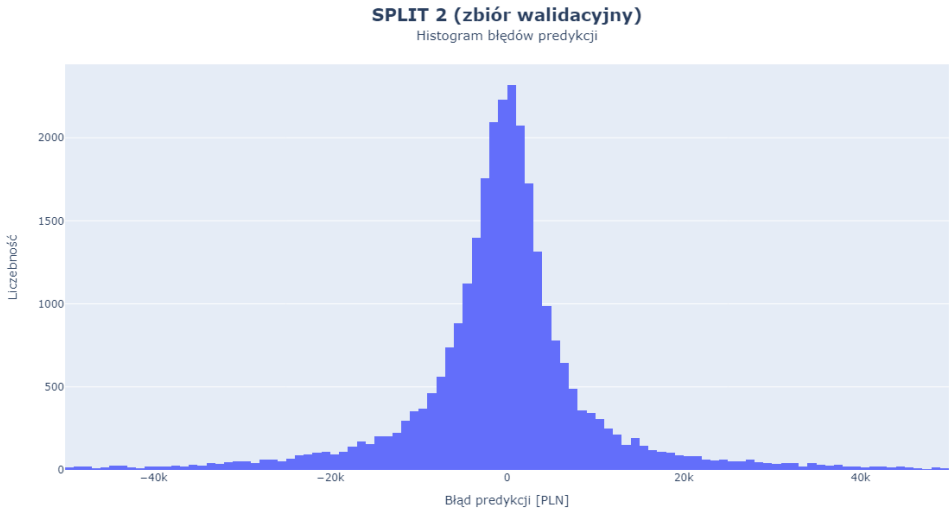


Rysunek 3.8: Histogram błędów predykcji na próbkach treningowych z zestawu danych SPLIT 2.

Na rys. 3.9 oraz rys. 3.10 przedstawiono wyniki ewaluacji modelu na zbiorze walidacyjnym. Skuteczność modelu pogorszyła się nieznacznie względem wyników uzyskanych na analogicznym podzbiorze z zestawu SPLIT 1 - średni błąd bezwzględny urósł do poziomu 9873 zł, a średni błąd procentowy zwiększył się o kilkanaście promili. Natomiast wartość metryki RMSE (oraz R^2) uległa drobnej poprawie, co świadczy o mniejszej liczbie popełnionych dużych błędów. Histogram błędów pozostał bez większych zmian.



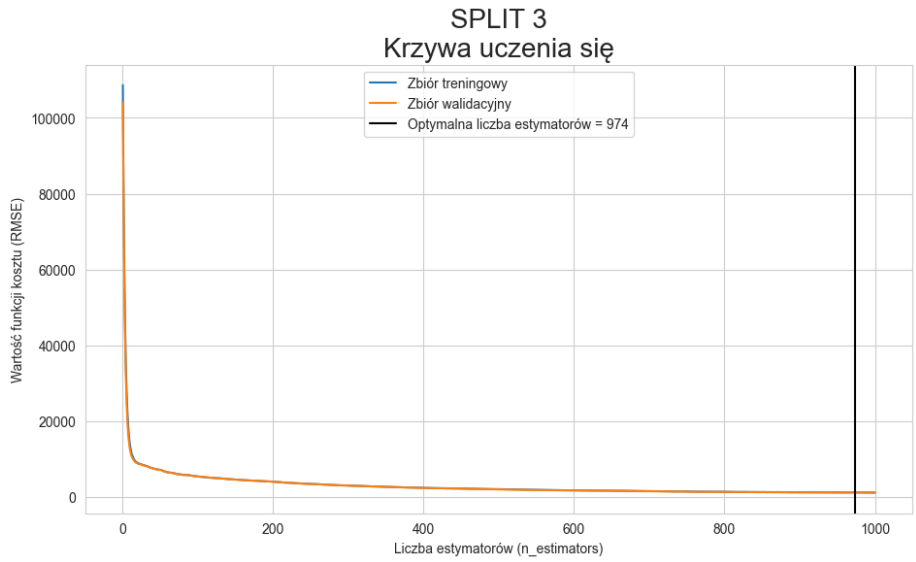
Rysunek 3.9: Porównanie cen rzeczywistych z predykcjami na próbkach walidacyjnych z zestawu danych SPLIT 2.



Rysunek 3.10: Histogram błędów predykcji na próbkach walidacyjncyh z zestawu danych SPLIT 2.

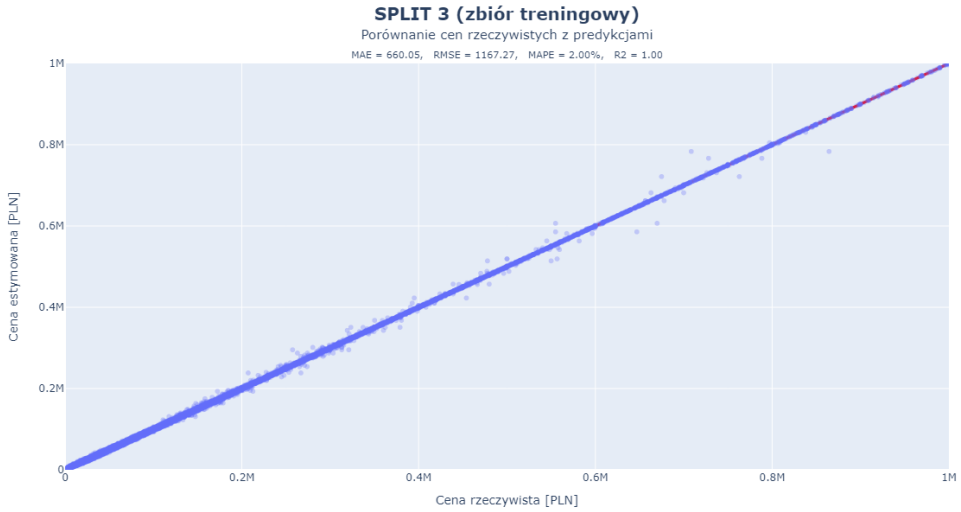
3.4. Trening modelu na zestawie danych SPLIT 3

Rysunek 3.11 przedstawia krzywe uczenia się modelu dopasowywanego do zestawu danych SPLIT 3 (MODEL 3). Wartości funkcji kosztu na zbiorze walidacyjnym oraz zbiorze treningowym maleją w identycznym tempie, ponieważ zbiór treningowy zawiera w sobie także zbiór walidacyjny zgodnie z postawionymi wymaganiami. W konsekwencji znaleziona optymalna liczba estymatorów jest znacznie większa niż poprzednio i wynosi 974. Trening modelu trwał aż 25 minut i 47 sekund właśnie z uwagi na niemalże trzykrotnie większą liczbę koniecznych do przeprowadzenia iteracji.

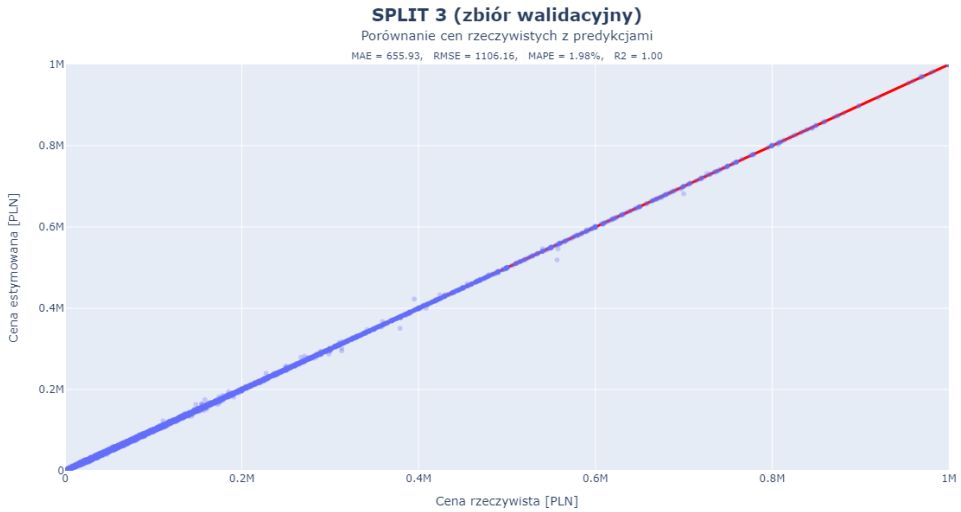


Rysunek 3.11: Krzywa uczenia się na zestawie danych SPLIT 3.

Na rys. 3.12 oraz rys. 3.13 przedstawiono wyniki ewaluacji modelu na zbiorach treningowym i walidacyjnym - są one niemalże identyczne z uwagi na fakt, że część walidacyjna stanowi podzbiór części treningowej. Dokładność estymacji modelu znacząco się poprawiła względem wyników uzyskanych na zbiorze treningowym w zestawie SPLIT 2 - średni błąd bezwzględny zmalał do poziomu 660 zł, a średni błąd procentowy wynosi teraz około 2%. Taka poprawa jest spowodowana dodaniem niemalże trzykrotnie większej liczby estymatorów do modelu, dzięki czemu był on w stanie jeszcze lepiej dopasować się do danych treningowych.

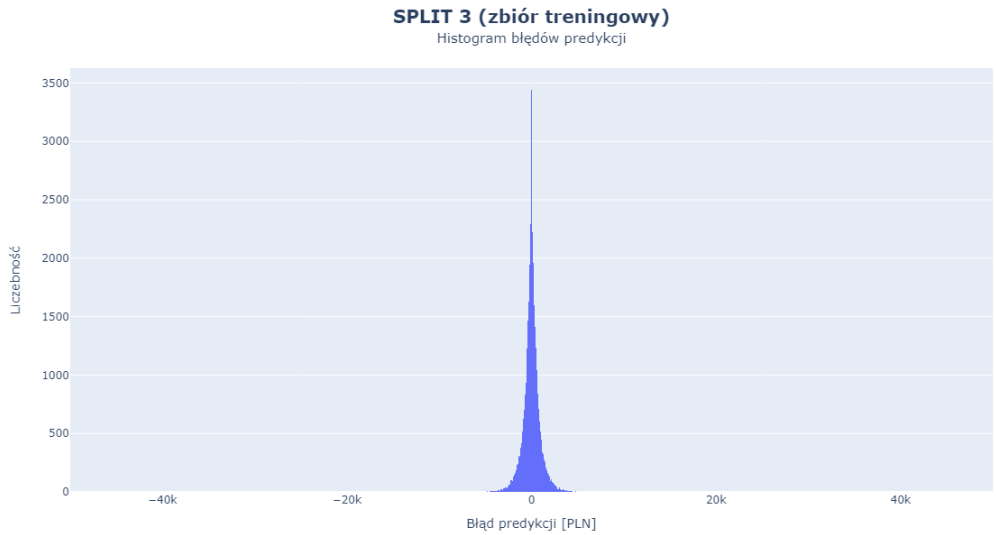


Rysunek 3.12: Porównanie cen rzeczywistych z predykcjami na próbkach treningowych z zestawu danych SPLIT 3.

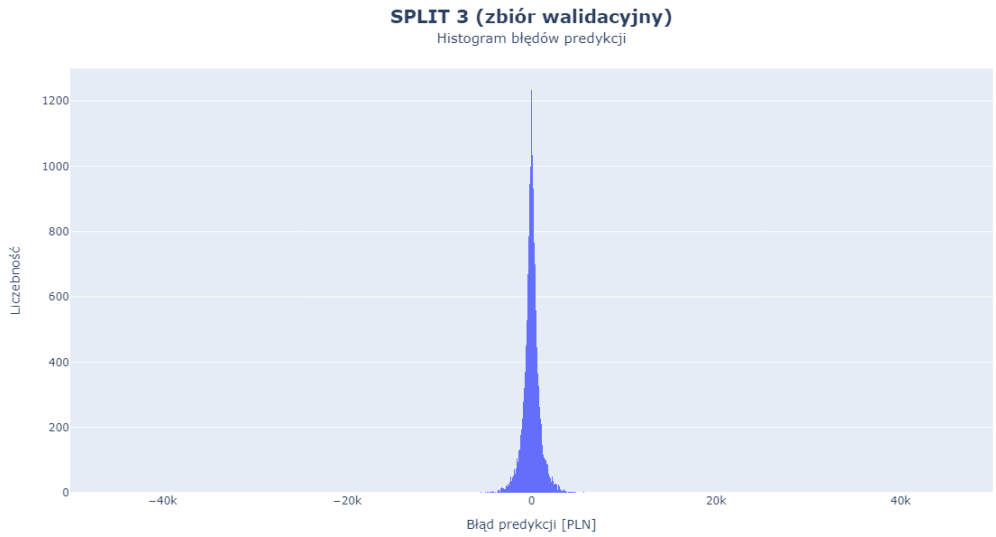


Rysunek 3.13: Porównanie cen rzeczywistych z predykcjami na próbkach walidacyjnych z zestawu danych SPLIT 3.

Z kolei na rys. 3.14 oraz rys. 3.15 przedstawiono histogramy błędów popełnionych przez model na zbiorach treningowym i walidacyjnym. Są one bardzo do siebie zbliżone z tego samego powodu, który opisano w poprzednim akapicie. Podobnie jak w przypadku pozostałych zestawów danych, tutaj także przypominają one rozkład normalny i są symetryczne względem 0. Jedyną różnicą jest ich zmniejszona szerokość (odchylenie standardowe błędów), co jest konsekwencją lepszego dopasowania modelu do danych treningowych.



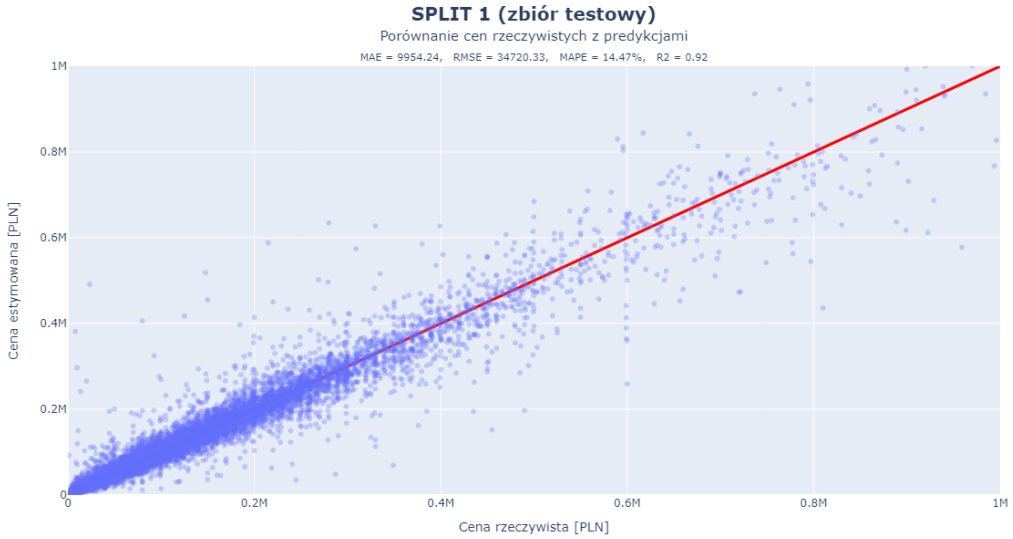
Rysunek 3.14: Histogram błędów predykcji na próbkach treningowych z zestawu danych SPLIT 3.



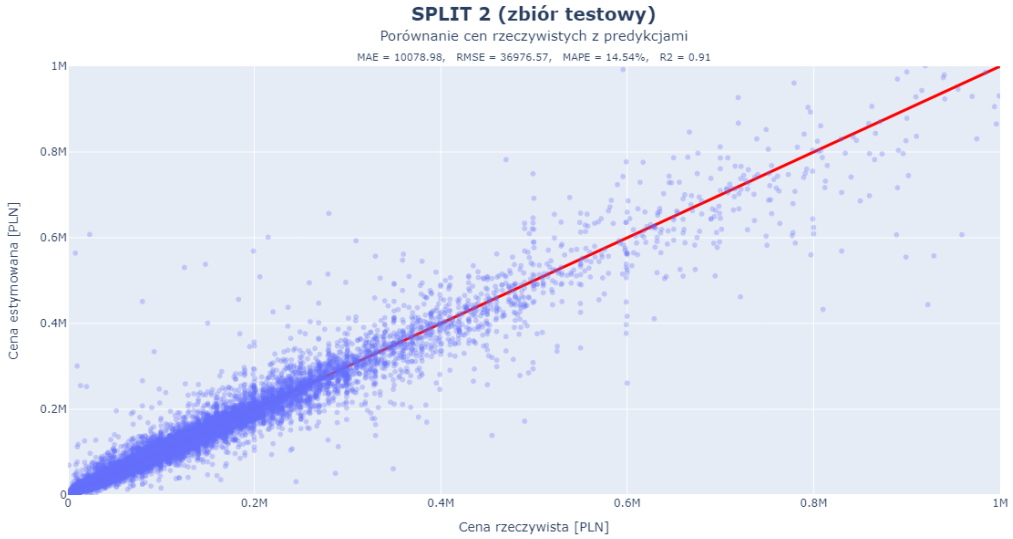
Rysunek 3.15: Histogram błędów predykcji na próbkach walidacyjncyh z zestawu danych SPLIT 3.

3.5. Testy modeli

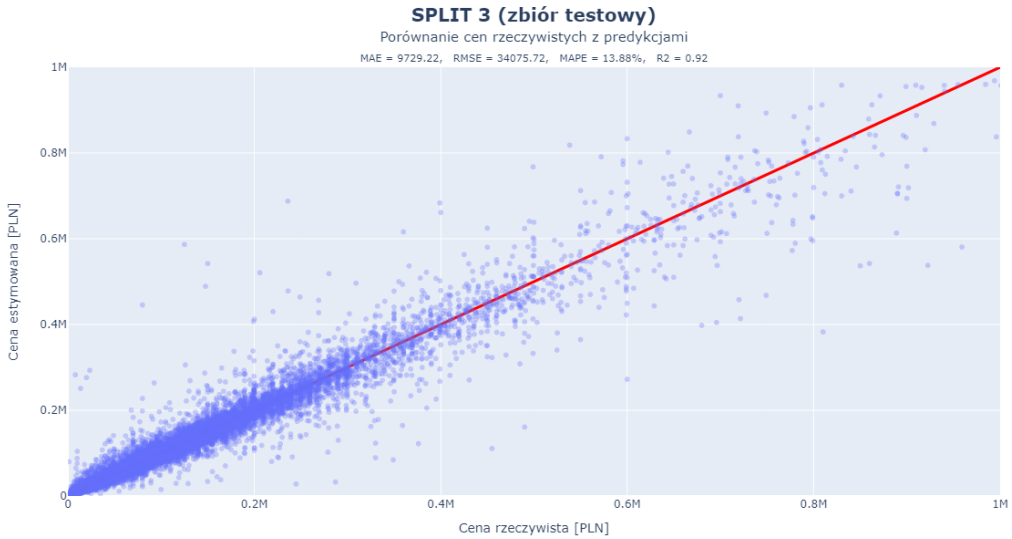
MODEL 1, MODEL 2 oraz MODEL 3 zostały także sprawdzone na zbiorach testowych z odpowiadających im zestawów danych (odpowiednio zestaw SPLIT 1, SPLIT 2 oraz SPLIT 3). Na rys. 3.16, rys. 3.17 oraz rys. 3.18 przedstawiono wyniki przeprowadzonych testów i uzyskane wartości metryk.



Rysunek 3.16: Porównanie cen rzeczywistych z predykcjami na próbkach treningowych z zestawu danych SPLIT 1.



Rysunek 3.17: Porównanie cen rzeczywistych z predykcjami na próbkach treningowych z zestawu danych SPLIT 2.



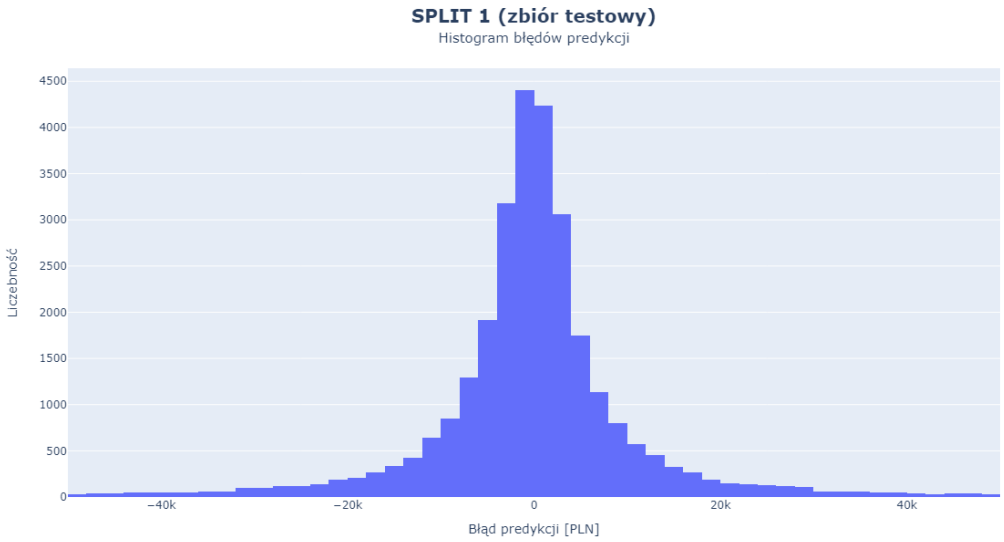
Rysunek 3.18: Porównanie cen rzeczywistych z predykcjami na próbkach treningowych z zestawu danych SPLIT 3.

Wyniki uzyskiwane przez wszystkie modele są bardzo do siebie zbliżone i nie różnią się zbytnio od wyników uzyskiwanych na zbiorach walidacyjnych (z wyjątkiem MODELU 3, gdzie zbiór walidacyjny był podzbiorem próbek treningowych). Ich estymacje różnią się średnio o około 10000 zł od cen rzeczywistych, co stanowi około 14% wartości pojazdów. Wartości współczynników R^2 wynoszą 0.92 lub 0.91 w zależności od modelu. Co zaskakujące, najlepsze wyniki uzyskał MODEL 3, podczas treningu którego do walidacji wykorzystano próbki treningowego, przez co dopasował się on w znacznie większym stopniu do podzbioru treningowego (ponad 3 razy większa liczba estymatorów). Najwidoczniej ryzyko przetrenowania zostało zniwelowane przez liczne mechanizmy regularyzacyjne, które są zaimplementowane w algorytmie XGBoost. Średni błąd procentowy tego modelu jest o ponad 0.5% mniejszy niż błędy dwóch pozostałych modeli. Z kolei wyniki MODELU 1 oraz MODELU 2 są niemalże identyczne, ponieważ operacje standaryzacji dotyczyły jedynie niewielkiego podzbioru cech wejściowych, przez co nie miało to wielkiego wpływu na jakość działania modelu. Dodatkowo, w odróżnieniu od modeli bazujących na obliczaniu odległości i/lub gradientów, drzewa decyzyjne są zdecydowanie mniej wrażliwe na skale cech wejściowych.

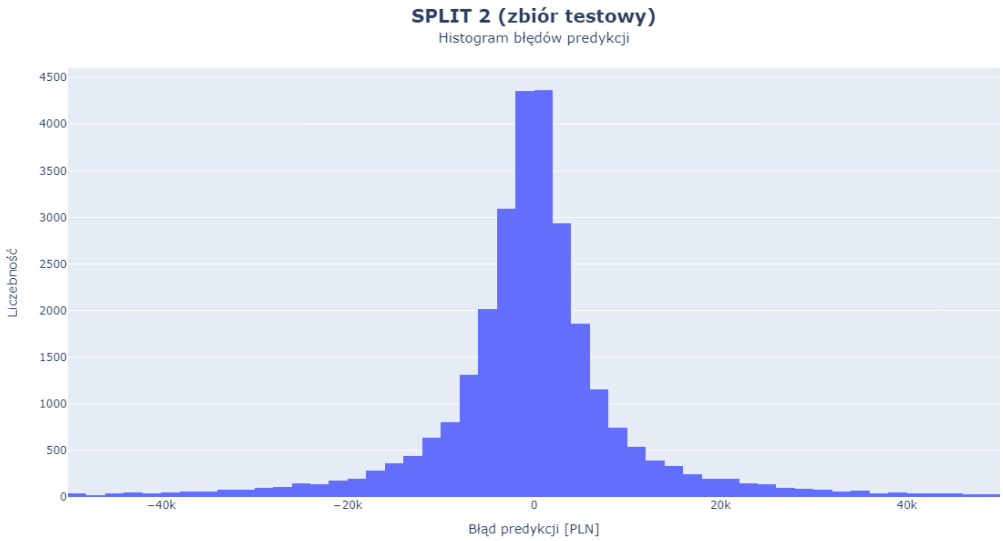
Na rys. 3.19, rys. 3.20 oraz rys. 3.21 wykreślono również histogramy błędów popełnianych przez modele - każdy z nich przypomina rozkład normalny i jest symetryczny względem wartości średniej, która leży w okolicach 0. Podobnie jak w przypadku testów na zbiorach walidacyjnych (z wyjątkiem zestawu SPLIT 3), tutaj także histogramy są nieco rozciągnięte na boki, ponieważ odchylenie standardowe błędów jest zdecydowanie większe niż w przypadku predykcji na zbiorach treningowych.

Sprawdzono także część przykładów testowych, dla których modele popełniały największe błędy estymacji. Zaobserwowane przyczyny pogorszonej jakości predykcji można podzielić na trzy następujące kategorie:

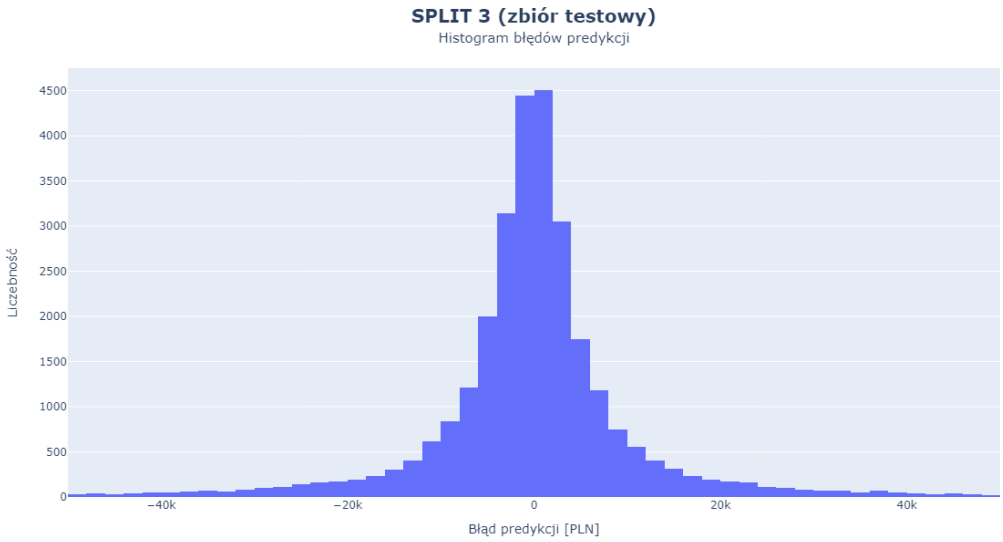
- Duże błędy estymacji dla samochodów drogich (cena » 300 tys. zł) wynikają głównie z niewielkiej liczby przykładów danego modelu w zebranych zbiorze danych.
- Część z grubych niedoszacowań modelu wynika z faktu uszkodzenia pojazdu, który nie został jednoznacznie określony przez sprzedającego w odpowiednim polu ogłoszenia.
- Część z grubych przeszacowań modelu wynika z faktu dodatkowego doinwestowania lub tuningu samochodów, który nie jest typowy dla danego modelu (np. drogie alufelgi lub wymiana silnika na mocniejszy).



Rysunek 3.19: Histogram błędów predykcji na próbkach testowych z zestawu danych SPLIT 1.



Rysunek 3.20: Histogram błędów predykcji na próbkach testowych z zestawu danych SPLIT 2.



Rysunek 3.21: Histogram błędów predykcji na próbkach testowych z zestawu danych SPLIT 3.

4. UDOSKONALENIE MODELU

Pomimo że wyniki osiągane przez modele przedstawione w Rozdział 3 są zadowalające, w tym rozdziale przestawione zostaną próby zwiększenia ich skuteczności poprzez zmiany przeprowadzone na etapie wstępnego przetwarzania danych oraz poprzez optymalizację hiperparametrów modelu.

4.1. Wstępne przetwarzanie danych

Zmiany na etapie wstępnego przetwarzania danych dotyczą głównie operacji uzupełniania brakujących wartości w kolumnach wejściowych. Dotychczasowa strategia polegała na uzupełnianiu braków medianą dla cech numerycznych oraz najczęściej występującą wartością dla cech kategorycznych.

W nowym podejściu cechy numeryczne nadal uzupełniane są medianą, jednak do każdej cechy dodawana jest także dodatkowa kolumna binarna, która przyjmuje wartości równe 1 dla przykładów, które początkowo posiadały brak wartości w kolumnie oryginalnej. Z kolei w przypadku cech kategorycznych, zamiast najczęściej występujących wartości, wszystkie braki zostały uzupełnione nową kategorią o nazwie ”Nieznany” w myśl zasady, że brak informacji to także informacja. Takie podejście może pozwolić modelowi wychwycić potencjalne sytuacje, w których sprzedający celowo nie zamieszczali konkretnej informacji, ponieważ mogłoby to obniżyć wartość pojazdu w oczach kupujących.

4.2. Optymalizacja hiperparametrów

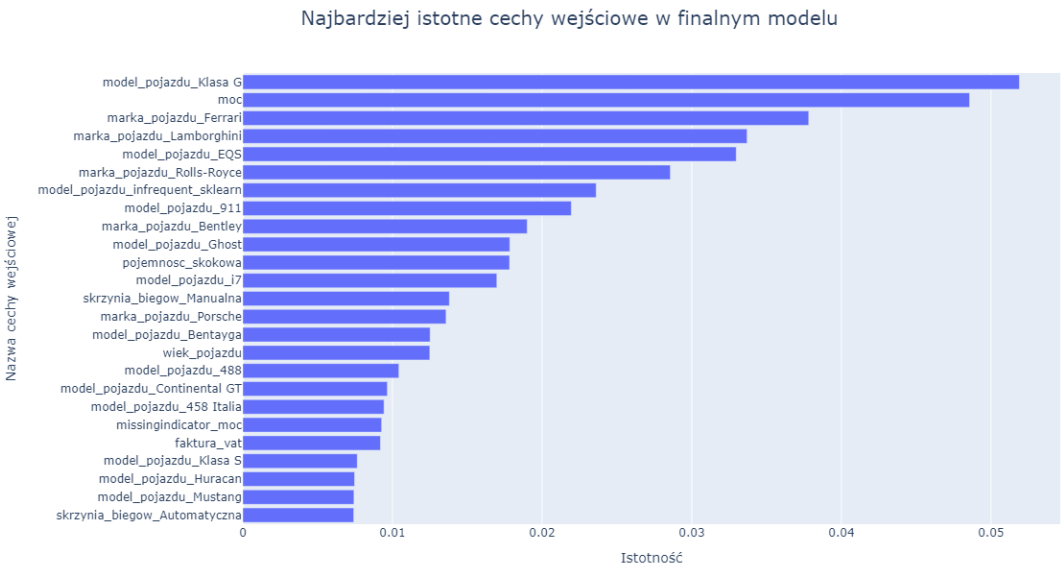
Poprzednio wartości wszystkich hiperparametrów (z wyjątkiem liczby estymatorów) zostały dobrane arbitralnie lub pozostawiono ich wartości domyślne. Takie podejście jest dalekie od optymalnego, ponieważ początkowe nastawy algorytmu XGBoost mają kluczowe znaczenie dla skuteczności jego działania i w znacznym stopniu zależą one od charakterystyki posiadanego zbioru danych oraz rozwiązywanego problemu. W konsekwencji niemożliwym jest wykorzystanie możliwości algorytmu XGBoost w pełni bez uprzedniego przeprowadzenia eksperymentów, mających na celu znalezienie jego najlepszych nastaw.

Do optymalizacji hiperparametrów wykorzystano pakiet Optuna, który zawiera efektywną implementację metody optymalizacji bayesowskiej, specjalnie dostosowaną do potrzeb tuningu modeli predykcyjnych. Działanie metody polega na próbach osza-

cowania początkowo nieznanej funkcji, której dziedziną jest podana przestrzeń hiperparametrów, a przeciwdziedziną przyjęta metryka błędu. Po znalezieniu jej przybliżenia zwracany taki zestaw wartości szukanych hiperparametrów, który będzie minimalizował błąd predykcji. Jest to metoda zdecydowanie bardziej efektywna niż przeszukiwanie siatki lub metody losowe.

Przeprowadzony eksperyment składał się z 250 iteracji trenowania modelu na zbiorze treningowym i testowania go na zbiorze walidacyjnym za pomocą metryki Mean Absolute Error. W ten sposób algorytm optymalizacji sprawdził skuteczność działania modelu przy różnych kombinacjach wartości dla jego najważniejszych parametrów początkowych. Do końcowych testów na podzbiorze testowym wybrano następujące wartości hiperparametrów:

- $n\_estimators = 999$
- $max\_depth = 17$
- $learning\_rate = 0.0326$
- $subsample = 0.6799$
- $colsample\_bytree = 0.6639$
- $gamma = 3.5521.$

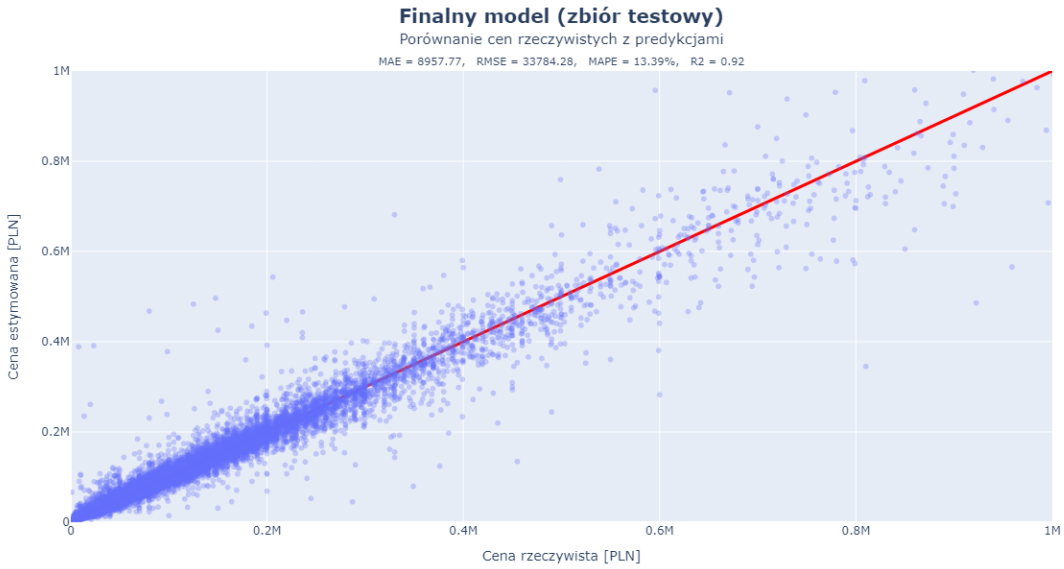


Rysunek 4.1: Wartości istotności dla cech najbardziej przydatnych podczas dopasowywania modelu.

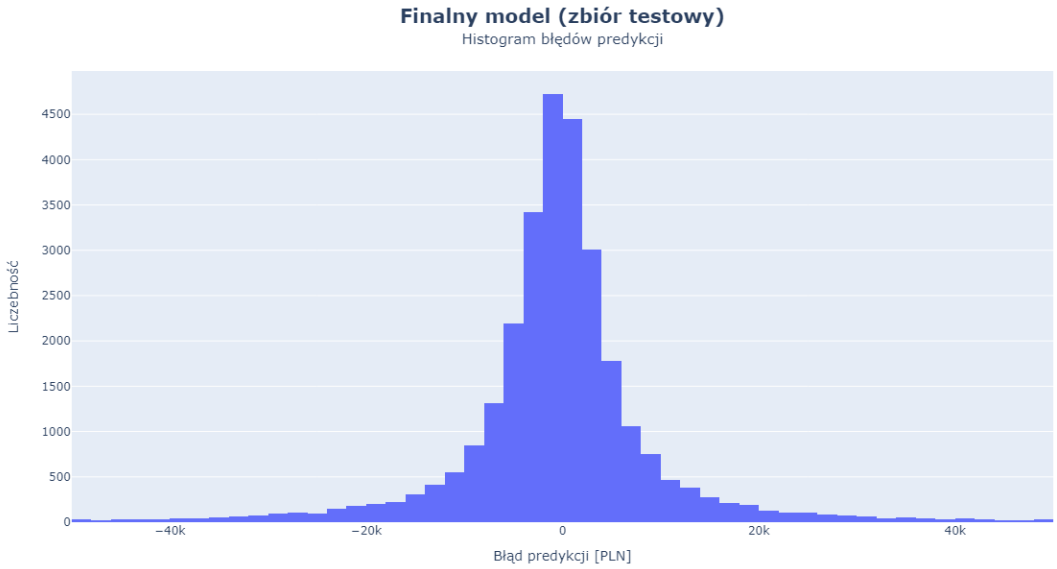
Dokonano także wglądu w najczęstsze cechy wybierane przez finalny model podczas generowania kolejnych drzew decyzyjnych. Nazwy najbardziej przydatnych kolumn wraz z obliczonymi wartościami ich istotności zostały przedstawione na rys. 4.1. Wyniki tego eksperymentu są zgodne z oczekiwaniami - zdecydowana większość najbardziej istotnych cech to zakodowane kolumny wskazujące na ekskluzywne marki i modele samochodów, których ceny są znacznie wyższe niż te przeciętne. Pozostałe cechy są także zgodne z intuicją - moc oraz pojemność silnika, wiek pojazdu, czy rodzaj skrzyni biegów to kluczowe czynniki wpływające na wartość pojazdu.

4.3. Sprawdzenie modelu na zbiorze testowym

Finalny model został sprawdzony na zbiorze testowym. Rysunek 4.2 przedstawia wyniki testów, na którym widzimy, że skuteczność modelu finalnego została poprawiona względem wyników uzyskiwanych przez poprzednie modele. Średni błąd bezwzględny udało się zredukować do poziomu poniżej 9 tys. zł, co stanowi około 13.5% wartości pojazdów. Stanowi to poprawę o ponad 1 tys. zł (metryka MAE) oraz 1.15 p.p. (metryka MAPE) względem MODELU 2. Współczynnik R^2 pozostał praktycznie bez zmian na poziomie 0.92. Wykreślono także histogram błędów predykcji, który jest widoczny na rys. 4.3. Nadal przypomina on rozkład normalny i jest symetryczny względem wartości średniej, która leży w okolicach 0.



Rysunek 4.2: Porównanie cen rzeczywistych z predykcjami finalnego modelu na próbkach testowych.



Rysunek 4.3: Histogram błędów predykcji modelu finalnego na próbkach testowych.

5. PODSUMOWANIE

Prace nad projektem rozpoczęto od napisania od podstaw programu dedykowanego do automatycznego zapisywania ofert sprzedażowych z portalu Otomoto.pl. W dalszej kolejności dokonano wstępnego czyszczenia danych oraz ekstrakcji istotnych cech. Następnie przeprowadzono obszerną analizę zebranego zbioru danych wraz z wizualizacją najistotniejszych statystyk. Kolejnym etapem był podział oryginalnego zbioru i przygotowanie trzech oddzielnych zestawów danych. Na każdym zestawie wytrenowano osobny model i osiągane przez nie wyniki porównano między sobą. Na sam koniec udoskonalono działanie systemu, zmieniając część z etapów pośrednich jego tworzenia oraz dokonując optymalizacji hiperparametrów.

Ostatecznie stworzono system, który z powodzeniem estymuje ceny samochodów osobowych, na podstawie podstawowych informacji powszechnie wykorzystywanych podczas transakcji kupna lub sprzedaży. Model popełniał błędy predykcji rzędu kilkunastu procent na zbiorze testowym składającym się prawie 30 tysięcy przykładów, co jest wynikiem co najmniej zadowalającym. Taka dokładność szacowania byłaby bardzo trudna do uzyskania dla osób niebędących ekspertami w dziedzinie motoryzacji. Świadczy to o potencjalnej możliwości wykorzystywania systemu jako pomocy w trakcie wstępnej oceny dotyczącej opłacalności zakupu danego pojazdu.