Analiza czynników wpływających na cenę samochodów

Weronika Biesalska

Warszawa 2024

Spis treści

1. Wstęp - Przygotowanie danych

W dzisiejszych czasach posiadanie samochodu stało się nieodłącznym elementem codziennego życia społecznego i funkcjonowania. Samochody nie są już tylko środkiem transportu, lecz również pełnią rolę wielofunkcyjnych narzędzi, wspierających nasze codzienne aktywności i komfort podróży. Zadaniem tego projektu jest dokładne zbadanie rynku samochodów osobowych i identyfikacja kluczowych czynników wpływających na ich cenę.

Celem moich badań jest zgłębienie wpływu różnorodnych determinantów na wartość rynkową samochodów osobowych oraz zrozumienie ich wzajemnych zależności. Chcę zdobyć głębszą wiedzę na temat kluczowych czynników, takich jak marka, model, przebieg czy stan techniczny, które decydują o cenach samochodów. Poniższe ustalenia mogą dostarczyć istotnych wskazówek i informacji, które pozwolą lepiej zrozumieć dynamiczny rynek samochodowy.

W rezultacie badania te mogą przyczynić się do zwiększenia efektywności strategii cenowych w branży samochodowej oraz poprawy sytuacji konsumentów poprzez dostarczanie konkurencyjnych cen i wysokiej jakości pojazdów, które spełniają różnorodne potrzeby użytkowników.

1.1. Opis zmiennych

Do projektu wykorzystałam zbiór danych "Vehicle Sales Cleaned" dostępny na platformie Kaggle (https://www.kaggle.com/datasets/krishanukalita/vehicle-sales-cleaned). Zawiera on szczegółowe informacje na temat 440 tysięcy różnych modeli samochodów, w tym ich cechy techniczne oraz etykiety cenowe. Badanie skupia się na tworzeniu modelu klasyfikacyjnego, który będzie prognozować klasę cenową samochodów na podstawie dostępnych zmiennych. Zestaw danych obejmuje 16 starannie wybranych zmiennych, odzwierciedlających różne techniczne aspekty i charakterystyki pojazdów, mogące wpływać na ich wartość rynkową. Poniżej opis zmiennych:

Tabela 1. Opis zmiennych

Nazwa zmiennej	Typ zmiennej	Opis/Wartości

Company	Zmienna nominalna (kategoryczna)	Producent samochodu, np. Ford, Chevrolet
Model	Zmienna nominalna (kategoryczna)	Konkretny model samochodu, np. F-150, Altima
Type	Zmienna nominalna (kategoryczna)	Typ pojazdu, np. Sedan, SUV
Size	Zmienna nominalna (kategoryczna)	Rozmiar samochodu, np. Compact, Mid-size
transmission	Zmienna nominalna (kategoryczna)	Rodzaj przekładni, np. Automatic, Manual
state	Zmienna nominalna (kategoryczna)	Stan, w którym samochód został sprzedany, np. ca (California), fl (Florida)
condition	Zmienna porządkowa	Ocena stanu technicznego samochodu, skala od 1 do 49
odometer	Zmienna ilościowa (liczbowa)	Przebieg samochodu w kilometrach. Zakres (1-999999)
color	Zmienna nominalna (kategoryczna)	Kolor zewnętrzny samochodu, np. Black, White.
interior	Zmienna nominalna (kategoryczna)	Kolor wnętrza samochodu, np. Black, Beige.
seller	Zmienna nominalna (kategoryczna)	Nazwa sprzedawcy samochodu, np. Kia Motors America Inc, Financial Services Remarketing.
mmr	Zmienna ilościowa (liczbowa)	Przybliżona wartość samochodu według określonej metody szacowania, w dolarach. Zakres (25-182000)

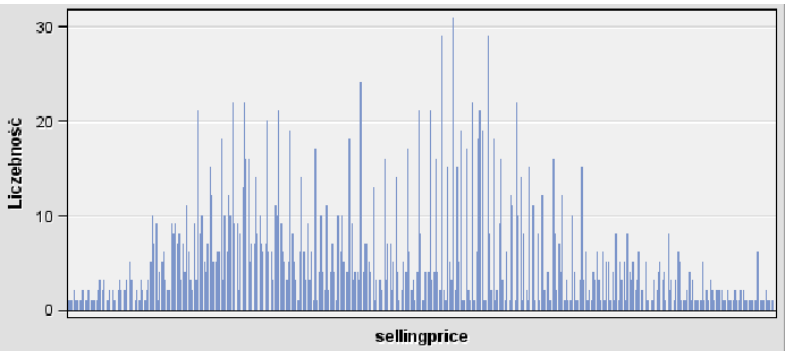
selling price	Zmienna ilościowa (liczbowa)	Rzeczywista cena sprzedaży samochodu, w dolarach (1-230000)
sale day	Zmienna nominalna (kategoryczna)	Dzień miesiąca, w którym odbyła się sprzedaż, wartości od 1 do 31.
sale month	Zmienna nominalna (kategoryczna)	Miesiąc, w którym odbyła się sprzedaż, np. Jan (January), Feb (February)
sale year	Zmienna nominalna (kategoryczna)	Rok, w którym odbyła się sprzedaż, np. 2014, 2015

Źródło: Opracowanie własne, https://www.kaggle.com/datasets/krishanukalita/vehicle-sales-cleaned

Po przeglądzie podstawowych statystyk zbioru danych usunęłam obserwacje z brakami w zmiennych.

Na tym etapie badania przeanalizowałam zmienną celu selling price, która reprezentuje rzeczywistą cenę sprzedaży samochodu. Aby uzyskać lepszy obraz rozkładu cen w moim zbiorze danych, stworzyłam histogram tej zmiennej.

Rysunek 1. Histogram zmiennej selling price



Źródło: Opracowanie własne

Ze względu na dużą liczbę rekordów w danych, sam histogram nie przedstawia w sposób jednoznaczny dominujących wartości. Dlatego też, aby ułatwić dalszą analizę i lepsze zrozumienie danych, zmienną sellingprice podzieliłam na pięć kategorii cenowych za pomocą nowej zmiennej price_category. Każda z tych kategorii obejmuje około 20% obserwacji, co umożliwia nam bardziej szczegółową analizę cech samochodów w różnych przedziałach

cenowych. Dzięki tej klasyfikacji możemy zauważyć, że rozkład sellingprice jest podobny dla każdej z tych kategorii.

Tabela 2. Rozkład liczebności zmiennej sellingprice stworzony przy pomocy nowej kategorii

price_category	Liczebność	Procent	Liczebność skumulowana	Procent skumulowany
Bardzo drogie	88021	19.99	88021	19.99
Bardzo tanie	88744	20.15	176765	40.14
Drogie	87643	19.90	264408	60.04
Tanie	88282	20.05	352690	80.09
Średnie	87703	19.91	440393	100.00

Źródło: Opracowanie własne

1.2. Odrzucenie zmiennych

Odrzucenie zmiennych w analizie danych jest zwykle podejmowane w celu zoptymalizowania procesu modelowania i poprawy jakości predykcji. W tym przypadku jednak postanowiłam nie odrzucać żadnych zmiennych ze zbioru danych. Ta decyzja oparta jest na przekonaniu, że każda zmienna wybranym zestawie danych może mieć istotny wpływ na przewidywaną cenę sprzedaży samochodu.

Przeprowadziłam dokładną analizę statystyczną i eksploracyjną danych, która nie dostarczyła silnych dowodów na nieistotność żadnej z zmiennych. W związku z tym, zachowałam wszystkie zmienne, aby pełniej reprezentować moje dane i uniknąć potencjalnej utraty istotnych informacji.

Takie podejście ma na celu maksymalne wykorzystanie wszystkich dostępnych danych, co może przyczynić się do lepszego zrozumienia zależności i predykcji cenowych w analizowanym zbiorze danych. Nie odrzuciłam żadnych zmiennych, aby móc skorzystać z ich potencjalnego wpływu na modelowanie i dalsze badania.

2. Drzewa decyzyjne

Analizowany zbiór danych składał się z około 440,000 obserwacji. Aby zapewnić właściwą ocenę modeli drzew decyzyjnych oraz ich zdolność do generalizacji, zbiór ten został podzielony na trzy części: zbiór uczący, zbiór walidacyjny oraz zbiór testowy. Podział został dokonany w proporcjach 40/30/30. Podział ten został wybrany, aby zapewnić optymalną jakość

modelu i jego zdolność do generalizacji. 40% danych przeznaczono na zbiór uczący, co daje modelowi wystarczająco dużo informacji do nauki i znalezienia wzorców w danych. 30% danych użyto jako zbiór walidacyjny, co pozwala na rzetelną ocenę wydajności modelu podczas dostrajania i uniknięcie przeuczenia. Pozostałe 30% danych przeznaczono na zbiór testowy, umożliwiając ostateczną ocenę zdolności modelu do przewidywania na nowych, nieznanych danych. Taki podział zapewnia równowagę między trenowaniem modelu a jego rzetelną oceną. Przed partycjonowaniem obserwacji, usunięto ze zbioru zmienną MMR, opisaną jako szacunkowa wartość rynkowa pojazdu. Zmienna ta miała podobne wartości do ceny samochodu, co mogłoby spowodować nadmierne dopasowanie.

2.1 Wybór najlepszego modelu

W celu wybrania najlepszego modelu drzewa decyzyjnego przewidującego kategorię cenową, przeprowadzono szeroką analizę różnych konfiguracji modeli. Analiza ta obejmowała testowanie kilku kluczowych parametrów, które mają wpływ na wydajność i dokładność modeli drzew decyzyjnych. Parametry te obejmowały:

- Maksymalne rozgałęzienie
- Maksymalna głębokość
- Minimalna wielkość zmiennej kategoryzującej
- Wielkość liścia

Każdy z tych parametrów został przetestowany w różnych konfiguracjach, a wyniki przedstawiono w formie tabeli, która zawierała współczynniki błędu klasyfikacji (MISC) dla zbiorów uczących, walidacyjnych oraz testowych. Współczynnik MISC jest kluczowym wskaźnikiem oceny jakości modelu, ponieważ mierzy stopień błędnych klasyfikacji, gdzie niższe wartości wskazują na lepszą wydajność modelu. Wyniki zebrano w poniższej tabeli.

Tabela 3. Wartości MISC dla poszczególnych iteracji drzewa decyzyjnego.

Max. rozgałęzienie	Max. głębokość	Min. wielkość zmiennej kategoryzującej	Wielkość liścia	MISC uczenie	MISC walidacja	MISC test
3	10	5	100	0,2426	0,2585	0,2603
3	7	5	100	0,2663	0,2758	0,2766
3	5	5	100	0,3248	0,3289	0,3303
4	7	5	44	0,2321	0,255	0,2524
5	7	5	44	0,2249	0,2514	0,251
5	10	5	44	0,2233	0,25	0,2502
3	4	5	44	0,381	0,3815	0,3826
5	6	5	44	0,2336	0,2563	0,2559
5	6	2	44	0,2254	0,2555	0,2547
5	7	2	44	0,2147	0,2508	0,2492
5	7	2	200	0,268	0,2787	0,2779
5	7	2	20	0,1963	0,2477	0,2466
5	7	2	30	0,2062	0,2479	0,2486
5	7	2	60	0,2247	0,2538	0,2542
5	7	2	80	0,233	0,2613	0,2597
5	7	2	40	0,2132	0,2513	0,2497
5	9	2	40	0,2108	0,2502	0,2487
5	7	4	40	0,2189	0,2489	0,2475
5	7	6	40	0,2252	0,2492	0,2497
5	6	4	40	0,228	0,254	0,2527
4	6	4	40	0,2452	0,2643	0,2644
5	5	5	40	0,2545	0,2713	0,2713

Na podstawie analizy wyników wybrano trzy modele, które charakteryzowały się najlepszymi wartościami współczynnika MISC dla zbiorów walidacyjnych i testowych. Modele te zostały zaznaczone na zielono w tabeli, co ułatwia ich identyfikację. Ostateczna selekcja najlepszego modelu opierała się na porównaniu tych trzech modeli pod kątem dodatkowych parametrów, aby zapewnić optymalną równowagę między dokładnością a zdolnością do generalizacji. Dokładniejsze wyniki zebrano w tabeli poniżej.

Tabela 4. Zestawienie 3 najlepszych modeli wraz z ich parametrami.

Wybrany model	Węzeł poprzedzający	Węzeł modelu	Opis modelu	Zmienna celu	Etykieta zmiennej celu	Kryterium wyboru: Valid: Misclassification Rate	Train: Sum of Frequencies	Train: Misclassification Rate	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error
Y	Tree3	Tree3	Maks roz...	price cat...		0.250814	176154	0.214699	0.999692	54609	0.062001	0.249001
	Tree4	Tree4	Maks roz...	price cat...		0.254015	176154	0.228062	0.99971	57524.22	0.065311	0.255561
	Tree2	Tree2	Maks roz...	price cat...		0.254984	176154	0.232087	0.999588	58438.51	0.066349	0.257584

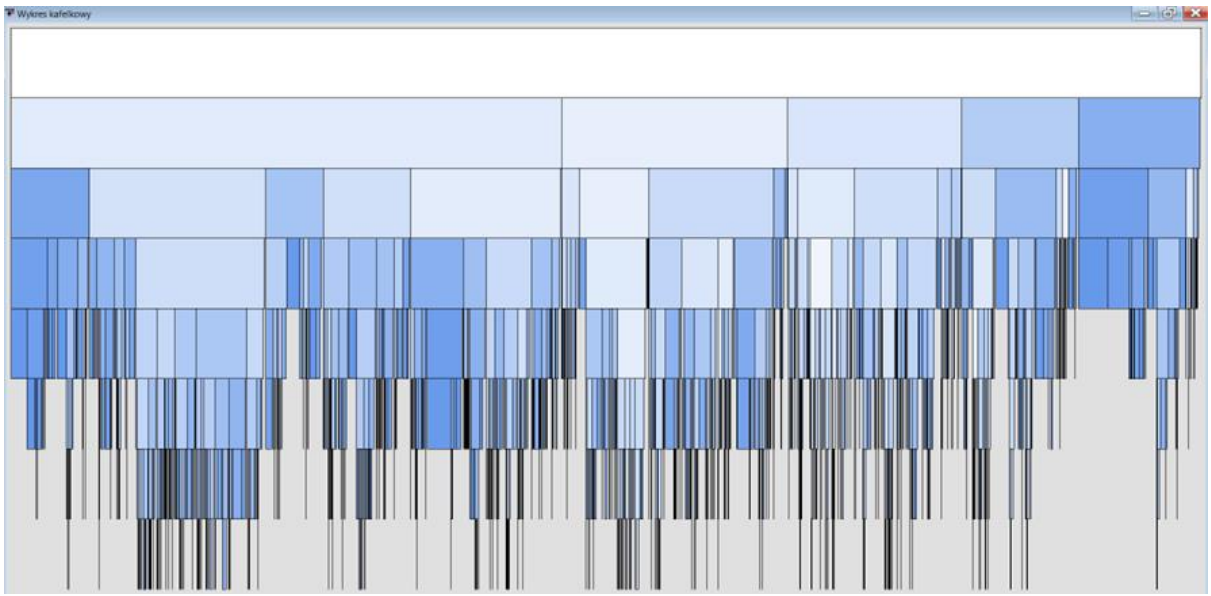
Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Frequencies	Valid: Misclassification Rate	Valid: Maximum Absolute Error	Valid: Sum of Squared Errors	Valid: Average Squared Error	Valid: Root Average Squared Error	Valid: Divisor for VASE	Test: Sum of Frequencies	Test: Sum of Weights Times Freqs	Test: Misclassification Rate
880770	704616	132118	0.250814	1	46933.55	0.071048	0.266548	660590	132121	660605	0.249173
880770	704616	132118	0.254015	1	47134.33	0.071352	0.267118	660590	132121	660605	0.2527
880770	704616	132118	0.254984	1	47274.78	0.071564	0.267515	660590	132121	660605	0.252428

Na podstawie analizy wyników, **Tree3** jest najlepszym wyborem. Model ten ma najniższy współczynnik błędu klasyfikacji na danych walidacyjnych i testowych oraz dobre wyniki w zakresie SSE, ASE i RASE, co wskazuje na jego wysoką dokładność i zdolność do generalizacji. Parametry modelu **Tree3** to:

- Maksymalne rozgałęzienie = 5
- Maksymalna głębokość = 7
- Minimalna wielkość zmiennej kategoryzującej = 2
- Wielkość liścia = 44

Wszystkie te czynniki razem wzięte – najniższy współczynnik błędu klasyfikacji, niskie wartości SSE, ASE i RASE – jednoznacznie wskazują, że **Tree3** jest najlepszym modelem spośród testowanych konfiguracji. Model ten charakteryzuje się wysoką dokładnością i zdolnością do generalizacji, co czyni go najbardziej odpowiednim wyborem do przewidywania kategorii cenowej w oparciu o dostępne dane. **Tree3** nie tylko minimalizuje błędy na zestawach walidacyjnych i testowych, ale również zapewnia stabilność i precyzję prognozowania, co jest kluczowe dla jego zastosowania w praktyce.

Rysunek 2. Wykres kafelkowy drzewa decyzyjnego.

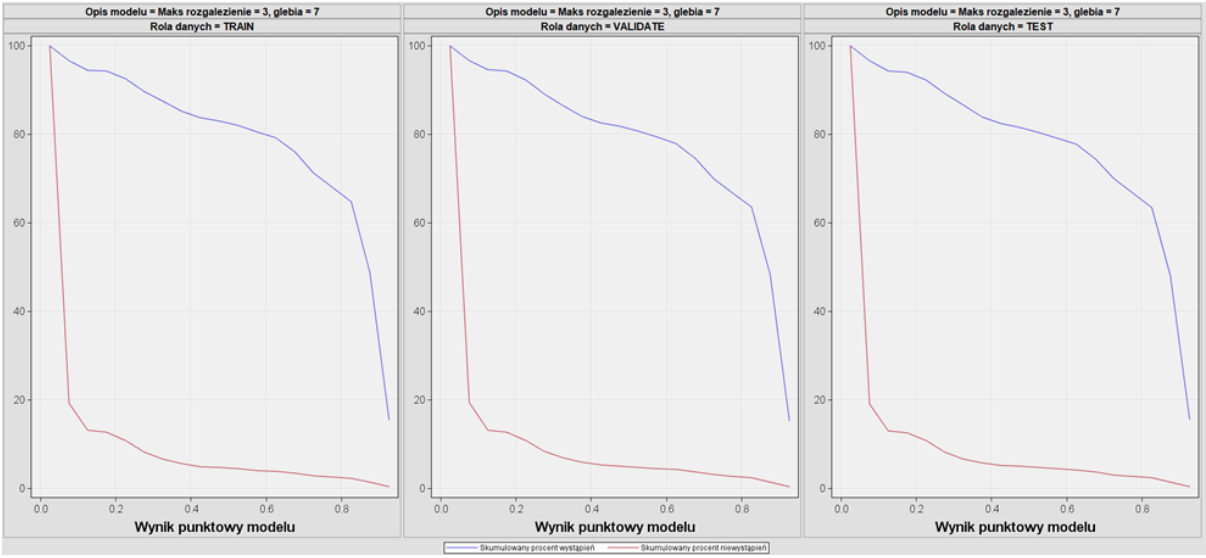


W trakcie modelowania próbowano różnych głębokości drzewa decyzyjnego, aby znaleźć najlepszą konfigurację. Testowano zarówno mniejsze głębokości (np. 3, 4), jak i większe (np. 9, 10). Okazało się jednak, że to właśnie głębokość 7 zapewniała najlepszą równowagę między dokładnością modelu a zdolnością do generalizacji. Większe modele z głębokością 9/10 powodowały zwiększenie różnicy pomiędzy zbiorem uczącym, a walidacyjnym. Modele te mimo większej głębokości nie klasyfikowały dużo lepiej wyników. Modele o mniejszej głębokości (4, 5) były zbyt uproszczone, co prowadziło do niedouczenia, gdzie model nie był w stanie uchwycić wszystkich istotnych wzorców w danych.

2.2 Krzywa ROC

Na przedstawionym poniżej rysunku 1 widzimy krzywe ROC dla trzech zestawów danych: uczącego (TRAIN), walidacyjnego (VALIDATE) i testowego (TEST). Każdy z wykresów przedstawia skumulowany procent wystąpień (true positive rate) oraz skumulowany procent niewystąpień (false positive rate) w zależności od wyniku punktowego modelu. Krzywe ROC dla modelu o maksymalnym rozgałęzieniu 3 i głębokości 7 pokazują, że model ten jest skuteczny w klasyfikacji danych na wszystkich trzech zestawach (uczących, walidacyjnych i testowych). Wysokie wartości AUC (Area Under Curve) oraz podobieństwo krzywych dla zbiorów walidacyjnych i testowych wskazują na dobrą zdolność do generalizacji i stabilność modelu. Model osiąga dość dobrą dokładność, skutecznie klasyfikując prawdziwe pozytywne przypadki przy minimalnej liczbie fałszywie pozytywnych klasyfikacji.

Rysunek 3. Krzywe ROC dla 3 zestawów danych.



- Krzywe ROC dla zbiorów walidacyjnych i testowych są podobne, co sugeruje, że model nie jest przeuczony
- Stabilne krzywe ROC na wszystkich trzech zestawach danych (uczących, walidacyjnych i testowych) świadczą o dobrze wytrenowanym modelu, który zachowuje swoją wydajność na różnych zestawach danych.

2.3 Istotność zmiennych

Tabela poniżej przedstawia wyniki analizy zmiennych użytych w modelu drzewa decyzyjnego. Zmienna *type (typ auta)* była używana 106 razy do podziału danych w drzewie, co daje jej istotność równą 0.6382. Zmienna ta była również często używana w kroswalidacji, z 1069 regułami, co odpowiada względnej ważności 0.6474. Istotność walidacji dla TYPE wynosi 0.6402, a stosunek istotności walidacji do istotności uczenia to 1.0031, co wskazuje na stabilność zmiennej między zbiorami uczącymi i walidacyjnymi.

Tabela 5. Istotność poszczególnych zmiennych.

Nazwa zmiennej	Etykieta	Liczba reguł podziału	Istotność	Liczba reguł dla drzew CV ▼	Ważność względna	Istotność walidacji	Iloraz istotności walidacji i istotności uczenia
TYPE		106	0.6382	1069	0.6474	0.6402	1.0031
odometer		89	1.0000	831	1.0000	1.0000	1.0000
seller		92	0.3996	814	0.4051	0.3830	0.9583
MODEL		68	0.9658	626	0.9980	0.9968	1.0321
condition		79	0.3003	624	0.3100	0.3089	1.0284
state		40	0.1324	390	0.1324	0.1278	0.9658
SIZE		10	0.4625	72	0.5377	0.5381	1.1633
COMPANY		7	0.6536	71	0.5810	0.5807	0.8885
Sale_month		7	0.0637	71	0.0730	0.0740	1.1627
interior		3	0.0374	38	0.0446	0.0454	1.2144
color		6	0.0316	23	0.0238	0.0287	0.9065
sale_Day		0	0.0000	6	0.0124	0.0165	.
transmission		0	0.0000	0	0.0000	0.0000	.
Sale_year		0	0.0000	0	0.0000	0.0000	.

Odometer (przebieg pojazdu) jest najważniejszą zmienną w modelu, z istotnością równą 1.0000. Była używana 89 razy do podziału danych i 831 razy w drzewach kroswalidacji. Wartość względna tej zmiennej to również 1.0000, a istotność walidacji wynosi 1.0000, co wskazuje na doskonałą stabilność (iloraz istotności walidacji i istotności uczenia wynosi 1.0000).

Zmienna *seller (sprzedawca pojazdu)* była używana 92 razy do podziału danych, osiągając istotność 0.3996. W drzewach kroswalidacji była używana 814 razy, co odpowiada względnej ważności 0.4051. Istotność walidacji dla seller wynosi 0.3830, a stosunek istotności walidacji do istotności uczenia to 0.9583, co wskazuje na pewne obniżenie znaczenia na zbiorze walidacyjnym w porównaniu do zbioru uczącego.

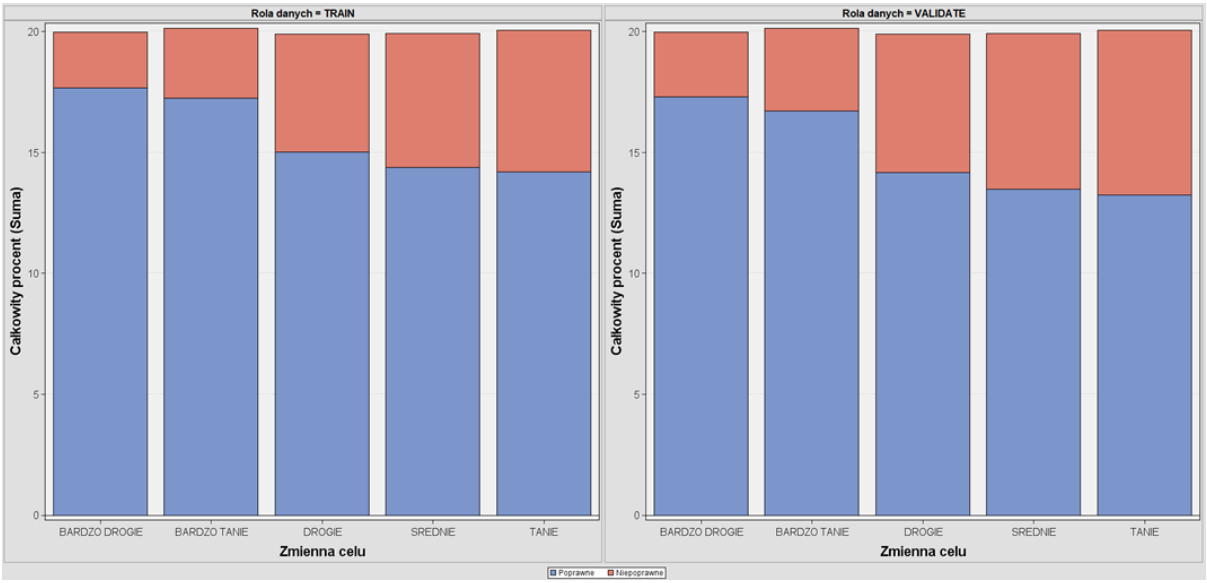
Zmienna *model* jest jedną z najważniejszych zmiennych, z istotnością równą 0.9658. Była używana 68 razy do podziału danych i 626 razy w kroswalidacji. Wartość względna tej zmiennej wynosi 0.9980, a istotność walidacji również wynosi 0.9980, co wskazuje na wysoką stabilność (iloraz istotności walidacji i istotności uczenia to 1.0231).

Condition (stan pojazdu) ma umiarkowaną istotność wynoszącą 0.3003. Była używana 79 razy do podziału danych i 624 razy w drzewach kroswalidacji. Wartość względna tej zmiennej to 0.3100, a istotność walidacji wynosi 0.3089, co wskazuje na dobrą stabilność zmiennej między zbiorami uczącymi i walidacyjnymi (iloraz istotności walidacji i istotności uczenia wynosi 1.0284).

Podsumowując, zmienne odometer, model, type, seler oraz condition są kluczowe dla tego modelu drzewa decyzyjnego, posiadając wysoką istotność i stabilność między zbiorami uczącymi i walidacyjnymi. Pozostałe zmienne takie jak company, sale month, interiori, color, sale day, transmission, sale year mają znacznie niższą istnotność oraz liczbę reguł.

2.4 Wykres klasyfikacji zmiennej celu price_category

Rysunek 4. Wykres klasyfikacji zmiennej celu.

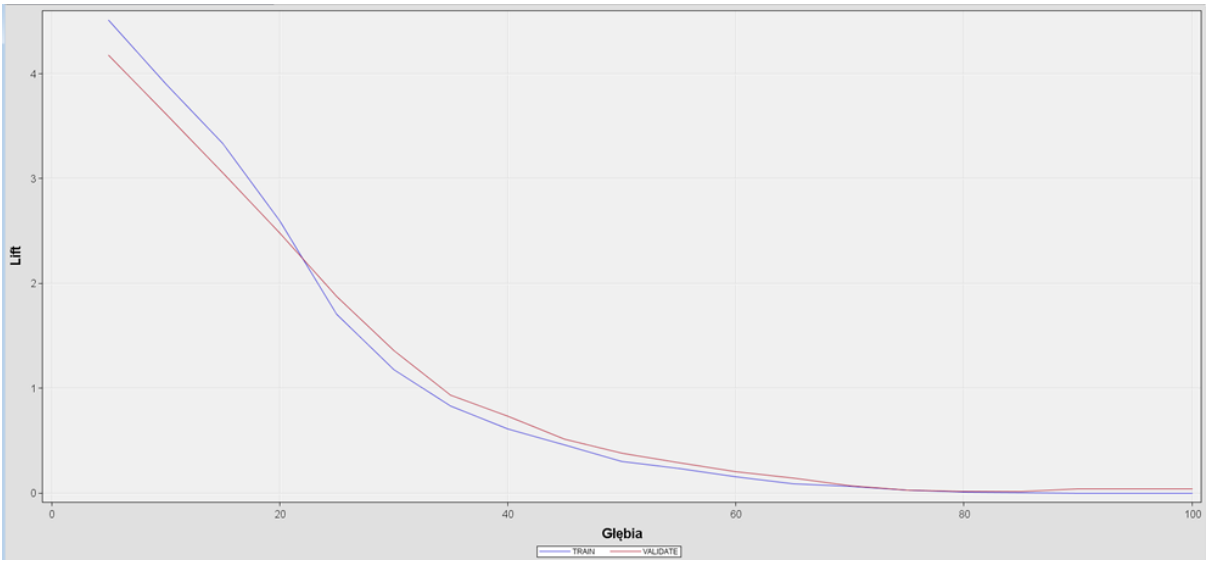


Na przedstawionym rysunku widzimy wykresy klasyfikacji dla drzewa decyzyjnego, które ilustrują, jak model przewiduje różne kategorie cenowe (BARDZO DROGIE, BARDZO TANIE, DROGIE, ŚREDNIE, TANIE). Wykresy te pokazują procentowy rozkład poprawnych (niebieskie) i niepoprawnych (czerwone) klasyfikacji dla zestawów danych uczących (TRAIN) i walidacyjnych (VALIDATE). Z podanego wykresu można wysunąć kilka wniosków:

- BARDZO DROGIE i BARDZO TANIE: Model osiąga wysoką skuteczność w klasyfikacji tych kategorii, z około 85-88% poprawnych klasyfikacji na obu zestawach danych. Oznacza to, że model jest szczególnie dobry w identyfikacji ekstremalnych kategorii cenowych
- DROGIE, ŚREDNIE i TANIE: Skuteczność modelu jest niższa dla tych kategorii, z około 70% poprawnych klasyfikacji. Model ma trudności z rozróżnianiem między bardziej zbliżonymi kategoriami cenowymi
- Wyniki dla zbioru uczącego i walidacyjnego są bardzo podobne, co wskazuje na dobrą zdolność modelu do generalizacji. Brak znaczących różnic między tymi zestawami sugeruje, że model nie jest przeuczony.

2.5 Wykres lift

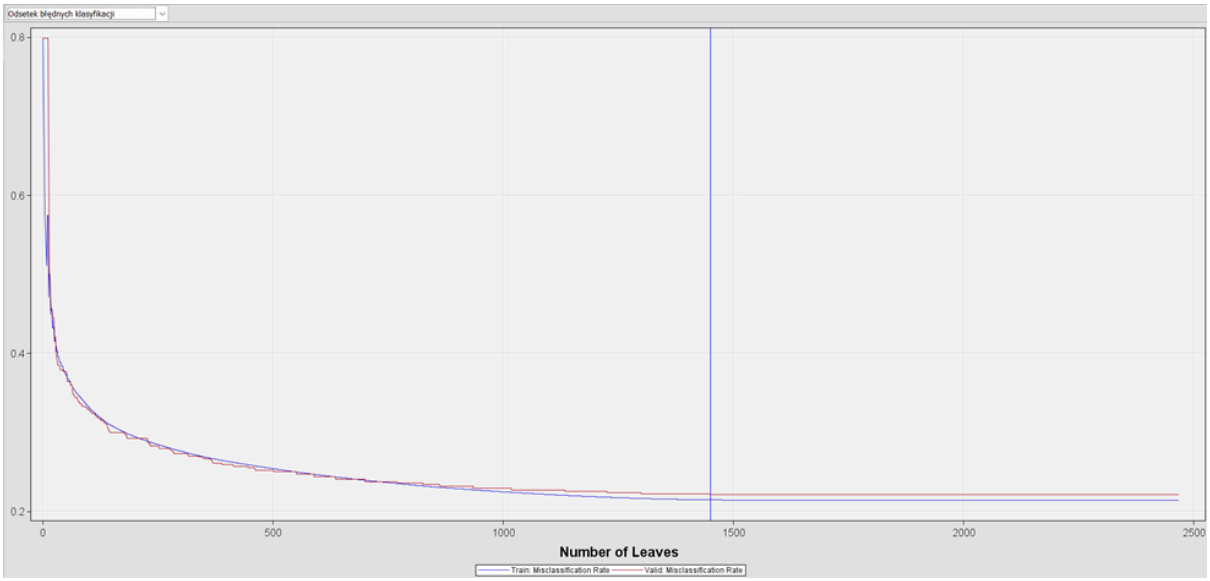
Rysunek 5. Wykres lift w zależności od głębi



Na przedstawionym wykresie widzimy krzywe liftu dla zestawów danych uczących (TRAIN) i walidacyjnych (VALIDATE). Lift jest mierzony w funkcji głębokości modelu drzewa decyzyjnego. Wykres pokazuje, jak zmienia się skuteczność modelu w identyfikacji pozytywnych przypadków w porównaniu do losowego zgadywania w zależności od głębokości drzewa. Wykres liftu pokazuje, że model drzewa decyzyjnego ma wysoką skuteczność w identyfikacji pozytywnych przypadków dla niskich wartości głębokości. W miarę zwiększania głębokości modelu, lift maleje, co sugeruje zmniejszenie skuteczności modelu. Mimo tego, bliskość krzywych dla zbiorów uczącego i walidacyjnego wskazuje, że model jest stabilny i ma dobrą zdolność do generalizacji. Model nie wykazuje znaczącego przeuczenia, co jest pozytywnym wskaźnikiem jego wydajności na nowych.

2.6 Wykres oceny poddrzewa

Rysunek 6. Odsetek błędnych klasyfikacji w zależności od liczby liści



Wykres przedstawia odsetek błędnych klasyfikacji (misclassification rate) w zależności od liczby liści (number of leaves) w drzewie decyzyjnym. Oś Y ilustruje odsetek błędnych klasyfikacji, gdzie niższe wartości oznaczają lepszą wydajność modelu, czyli mniejszą liczbę błędnych klasyfikacji. Oś X przedstawia liczbę liści w drzewie decyzyjnym, które są końcowymi węzłami drzewa reprezentującymi ostateczne decyzje klasyfikacyjne. Większa liczba liści wskazuje na bardziej szczegółowe drzewo z większą liczbą decyzji końcowych. Na wykresie widzimy dwie krzywe: niebieską linię reprezentującą odsetek błędnych klasyfikacji dla zbioru uczącego (Train: Misclassification Rate) oraz czerwoną linię dla zbioru walidacyjnego (Valid: Misclassification Rate). Na początku, gdy liczba liści jest mała, odsetek błędnych klasyfikacji jest wysoki, co oznacza, że model jest zbyt prosty i nie potrafi dobrze klasyfikować danych. W miarę zwiększania liczby liści, odsetek błędnych klasyfikacji maleje, co wskazuje, że model staje się bardziej złożony i lepiej dopasowuje się do danych uczących. Krzywe dla zbiorów uczącego i walidacyjnego są do siebie bardzo zbliżone, co sugeruje, że model dobrze generalizuje i nie jest przeuczony.

Po osiągnięciu pewnej liczby liści, około 1500 (niebieska linia odpowiada wybranemu modelowi), odsetek błędnych klasyfikacji stabilizuje się i nie zmienia się znacząco wraz ze wzrostem liczby liści. Punkt ten może wskazywać na optymalną liczbę liści, po której dalsze zwiększanie złożoności modelu nie przynosi już znaczącej poprawy wydajności. Stabilizacja

krzywych na niskim poziomie błędu klasyfikacji sugeruje, że model osiągnął swoje maksimum zdolności predykcyjnej.

Podsumowując, wykres ten pokazuje, jak złożoność modelu wpływa na jego wydajność i pozwala na identyfikację punktu, w którym dalsza złożoność nie przynosi korzyści. Bliskość krzywych dla zbiorów uczącego i walidacyjnego wskazuje na stabilność i dobrą zdolność do generalizacji modelu, co jest pozytywnym sygnałem dla jego użyteczności w praktyce.

2.7 Drzewo decyzyjne bez zmiennych z istotnością = 0

W celu zoptymalizowania modelu drzewa decyzyjnego przewidującego kategorię cenową, przeprowadzono analizę istotności zmiennych użytych w modelu. Tabela istotności zmiennych wykazała, że niektóre zmienne nie miały żadnego wpływu na podziały w drzewie decyzyjnym, co oznacza, że ich istotność wynosiła 0. Zmienne te to: *sale Day, transmission* oraz *Sale year*. Dążąc do uproszczenia modelu i poprawy jego wydajności operacyjnej, postanowiono stworzyć nowe drzewo decyzyjne z wykluczeniem tych zmiennych. Celem tej analizy jest ocena, czy usunięcie zmiennych o zerowej istotności wpłynie pozytywnie na działanie modelu, jednocześnie zachowując jego dokładność i zdolność do generalizacji. Porównamy wyniki nowego modelu z obecnym, aby upewnić się, że uproszczenie nie prowadzi do pogorszenia jakości predykcji. Poniżej w tabeli zaprezentowano otrzymane wyniki.

Tabela 6. Statystyki dopasowania dla modelu bez zmiennych z istotnością = 0

Zmienna celu	Etykieta zmiennej celu	Statystyki dopasowania	Etykieta statystyk	Uczenie	Walidacja	Test
price category		NOBS	Sum of Frequencies	176154	132118	132121
price category		MISC	Misclassification ...	0.214699	0.250814	0.249173
price category		MAX	Maximum Absolut...	0.999692	1	1
price category		SSE	Sum of Squared E...	54609	46933.55	46801.15
price category		ASE	Average Squared ...	0.062001	0.071048	0.070846
price category		RASE	Root Average Squ...	0.249001	0.266548	0.266169
price category		DIV	Divisor for ASE	880770	660590	660605
price category		DFT	Total Degrees of ...	704616	.	.

Wyniki tej analizy wskazują, że usunięcie tych zmiennych nie przyniosło ani poprawy, ani pogorszenia wydajności modelu. Odsetek błędnych klasyfikacji pozostał na podobnym poziomie, a zdolność modelu do generalizacji nie uległa znaczącej zmianie. Warto jednak zachować te zmienne, ponieważ:

- mogą okazać się wartościowe w przyszłych modelach lub innych technikach analitycznych. Zachowanie ich w zbiorze danych umożliwia ich łatwe wykorzystanie, jeśli okaże się, że w innej konfiguracji modelu mają większe znaczenie
- mogą mieć znaczenie z punktu widzenia biznesowego lub operacyjnego. Mogą dostarczać dodatkowych informacji, które są istotne dla interpretacji wyników lub podejmowania decyzji
- pozwala to na utrzymanie pełnego zestawu zmiennych zapewnia kompletność danych.

3. Lasy losowe

Lasy losowe to zaawansowana technika uczenia maszynowego, która opiera się na konstrukcji wielu drzew decyzyjnych i łączeniu ich wyników w celu klasyfikacji lub regresji. Każde drzewo jest trenowane na podstawie losowego próbkowania z powtórzeniami ze zbioru treningowego, co pozwala uniknąć przetrenowania i wprowadza różnorodność w strukturze modelu. Ostateczna decyzja klasyfikacyjna lub predykcja regresyjna jest podejmowana na podstawie głosowania większościowego nad klasami wskazanymi przez poszczególne drzewa decyzyjne.

Lasy losowe charakteryzują się odpornością na przetrenowanie, efektywnością w pracy z dużymi zbiorami cech oraz wysoką zdolnością do generalizacji. Dzięki swojej elastyczności i wysokiej wydajności, są jednym z najbardziej popularnych i użytecznych algorytmów w dziedzinie uczenia maszynowego, zapewniając precyzyjne predykcje i łatwą interpretację wyników.

W analizie zastosowano model do przewidywania rzeczywistej ceny sprzedaży samochodów, reprezentowanej przez zmienną sellingprice, której wartości zostały podzielone na pięć kategorii cenowych w zmiennej price_category.

Z analizowanego zbioru danych usunięto zmienną MMR, która reprezentowała szacowaną wartość rynkową pojazdu. Wartości zmiennej były zbliżone do rzeczywistych cen samochodów, co mogło prowadzić do zbyt silnego dopasowania modelu do danych treningowych. Następnie, wybrano próbkę 10% danych z całego zbioru, aby zbalansować efektywność obliczeniową z dokładnością modelowania. Dane podzielono na zbiór treningowy i testowy w proporcji 80/20.

3.1. Skuteczność Modelu

Model lasów losowych zastosowany do przewidywania kategorii cenowych samochodów wykazał umiarkowaną skuteczność, uzyskując średnią dokładność na poziomie 70.08%. Wyniki walidacji krzyżowej, z dokładnościami kolejno: 70.65%, 69.96%, 70.59%, 71.07%, 70.90%, ze średnią dokładnością 70.63%, są spójne z dokładnością uzyskaną na zbiorze testowym. Stabilność wyników walidacji krzyżowej potwierdza, że model zachowuje podobną skuteczność niezależnie od tego, jakie dane są używane do testowania, co jest pozytywnym wskaźnikiem jego generalizacji oraz odporności na przeuczenie.

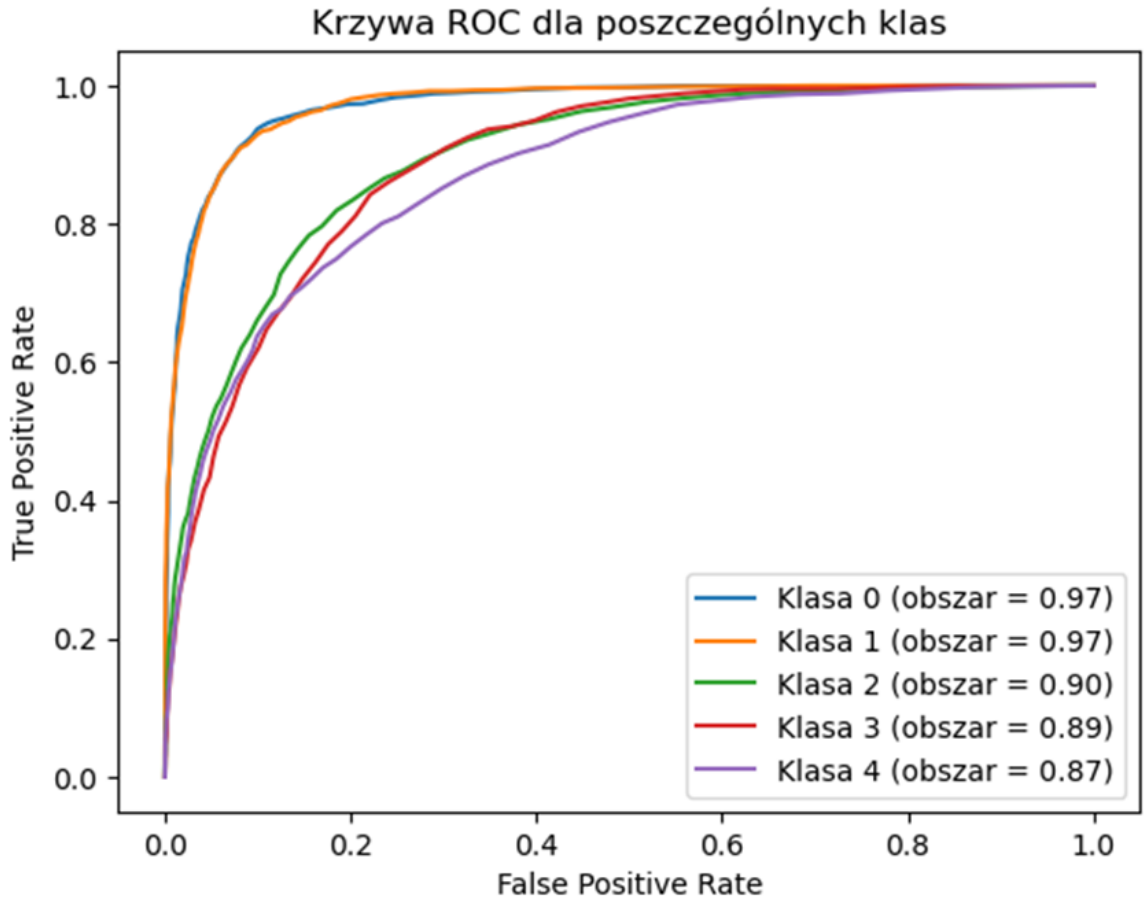
3.2. Krzywa ROC

Krzywa ROC ilustruje skuteczność modelu lasów losowych w rozróżnianiu pomiędzy różnymi kategoriami cenowymi samochodów, reprezentowanymi przez pięć klas (0 - 'Bardzo drogie', 1 - 'Bardzo tanie', 2 - 'Drogie', 3 - 'Tanie', 4 - 'Średnie'). Analizując krzywą, można dokonać następujących interpretacji:

- Wysoki wynik AUC dla klasy 0 oraz 1 wynoszący 0.97 wskazuje, że model skutecznie identyfikuje obserwacje tej klasy z minimalnym błędem. Niska wartość FPR dla tej klasy oznacza, że model rzadko mylnie identyfikuje inne klasy jako klasę 0 lub 1. Wysoki TPR pokazuje, że większość prawdziwych klas 0 oraz 1 jest poprawnie klasyfikowana.
- AUC dla klasy 2 wynosi 0.90. Choć jest on nieco niższy niż dla klas 0 i 1, to nadal jest dobry. Wskazuje on na zdolność modelu do stosunkowo skutecznego rozróżniania tej klasy. Warto zwrócić uwagę na możliwe przyczyny niższego AUC, takie jak mniejsze różnice cech między tą a innymi klasami.
- Wynik AUC dla klasy 3 wynosi 0.89, co wskazuje na to, że model może mieć lekkie trudności z dokładnym klasyfikowaniem tej kategorii. Możliwe, że klasa ta jest bardziej podobna do innych klas, co sprawia, że model częściej popełnia błędy.
- Wynik AUC dla klasy 4 (0.87) jest najniższym spośród wszystkich klas, co może sugerować, że ta kategoria cenowa jest najtrudniejsza do rozróżnienia. Może to wynikać z cech wspólnych tej kategorii z innymi klasami lub z niewystarczającej liczby przykładów tej klasy w danych treningowych.

Ogólnie rzecz biorąc, model wykazuje umiarkowaną dokładność i dobre zdolności rozróżniania klas, co czyni go przydatnym narzędziem, ale wskazuje również na możliwość dalszej optymalizacji..

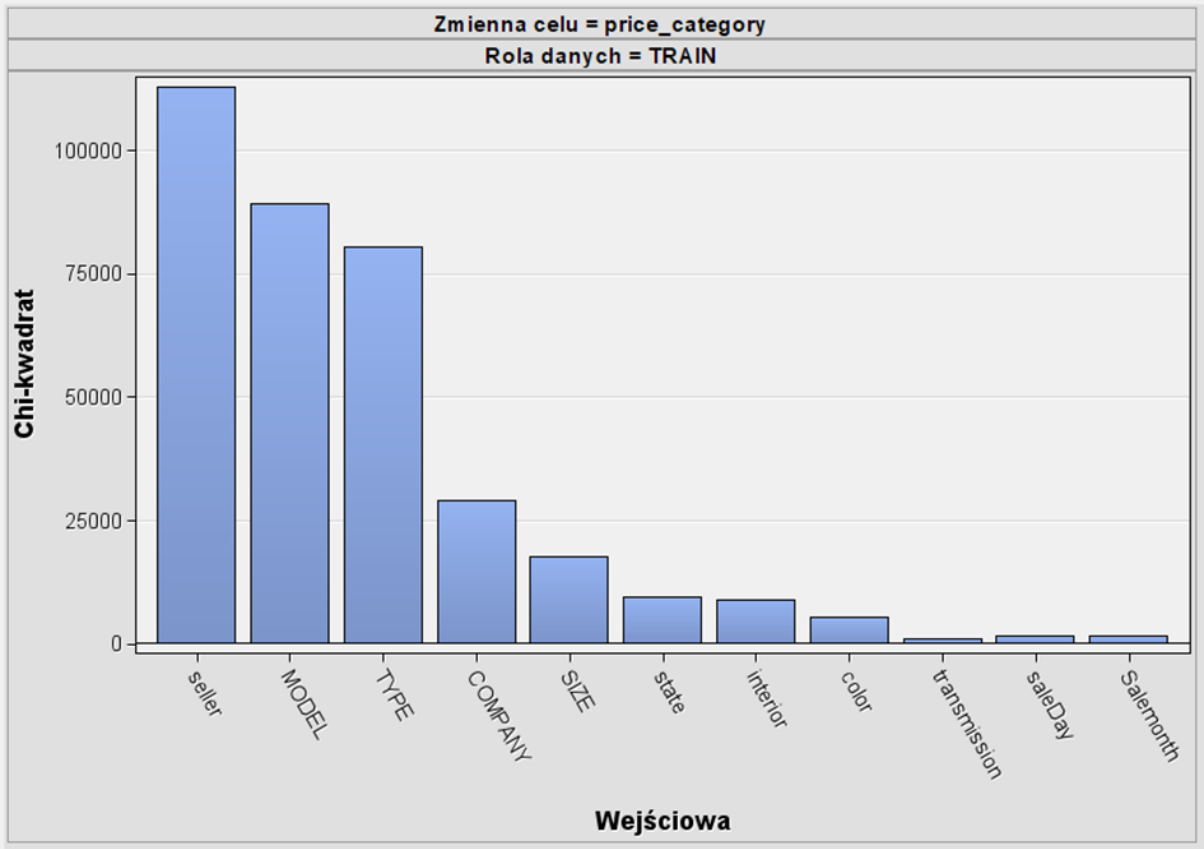
Rysunek 7. Krzywa ROC dla poszczególnych klas



4. Sieci neuronowe

W celu zbadania siły związku zmiennych na zmienną celu użyto węzła *„Eksploatacja statystyk”*. Wyniki zamieszczono poniżej[Rysunek 7.].

Rysunek 8. Statystyka Chi-kwadrat dla zbioru danych



Jak można zauważyć, według statystyki Chi-kwadrat najmniejszy wpływ na zmienną celu mają zmienne: transmission, saleDay, Salemonth. Statystyka ta się przyda się w późniejszym tworzeniu jednego z wariantów sieci neuronowej.

4.1 Budowa modeli sieci neuronowych

Zbiór danych został podzielony na zbiory: uczący(60%), walidacyjny(30%), testowy(10%).

Pierwszy model sieci nie powiódł się z powodu ograniczeń sprzętowych, w związku z dużą liczbą zmiennych i wielkości zbioru. Model ten posiadał ustawienia domyślne, zmieniono jedynie parametr „Maksymalnie iteracji” z 50 na 500, w celu osiągnięcia zbieżności.

Rysunek 9. Pierwszy model

Aby zmniejszyć zbiór użyto węzła *„Próbkowanie”* w celu zmniejszenia. Wiersze będą wybieranie metoda domyślną oraz wielkość zbioru analizowanego będzie 10% zbioru wejściowego.

Rysunek 10. Drugi model

Drugi model też nie przeliczył się, z powodu ograniczeń.

Rysunek 11. Błąd - za duża liczba zmiennych

```
ERROR: W pliku EMWS2.NEURAL4_INITIAL zdefiniowano zbyt wiele zmiennych. Plik nie może mieć więcej niż 17905 zmiennych.
```

W celu porzucenia zmiennych o małej częstości występowania użyto węzła *„Filtrowanie”*. Węzeł usuwa zmienne o częstości występowania poniżej 1%.

Rysunek 12. Trzeci model

Sieć neuronowa(5) została uruchomiona ze standardowymi ustawieniami oraz ze zwiększoną liczbą iteracji.

Tabela 7. Rezultaty dla pierwszego podejścia do modelowania sieci neuronowej

	Odsetek błędnych klasyfikacji	Przeciętny błąd kwadratowy

Numer sieci	Uczenie	Walidacja	Test	Uczenie	Walidacja	Test
5	0.274514	0.36904	0.367679	0.069618	0.0887	0.087287

Sieci neuronowe 6 i 7 miały ustawione parametry: „Kryterium wyboru modelu" na „Średni błąd" oraz „Liczba jednostek ukrytych" odpowiednio na 1 i 4. Rezultaty wyglądają następująco:

Tabela 8. Rezultaty dla sieci neuronowych i różną liczbą jednostek ukrytych

Numer sieci	Odsetek błędnych klasyfikacji			Przeciętny błąd kwadratowy		
	Uczenie	Walidacja	Test	Uczenie	Walidacja	Test
6	0.304273	0.386379	0.383091	0.080363	0.094716	0.093998
7	0.262096	0.264117	0.358432	0.065802	0.087877	0.086259

W następnym kroku użyto węzła *„Wybór zmiennych"*, który odrzuca zmienne niespełniające warunków.

Rysunek 13. Gałąź z węzłem dla czwartego podejścia do modelowania sieci

I tak poniżej [Rysunek 14.] pokazuje zmienne przepuszczone do dalszej analizy:

Rysunek 14. Zmienne przepuszczone przez węzeł do dalszej budowy modelu sieci neuronowej

Nazwa	Użycie	Raport	Rola	Poziom
COMPANY	**Domyślne**	**Nie**	Wejście	Nominalna
SIZE	**Domyślne**	**Nie**	Wejście	Nominalna
condition	**Domyślne**	**Nie**	Wejście	Przedziałowa
odometer	**Domyślne**	**Nie**	Wejście	Przedziałowa
price_category	**Tak**	**Nie**	Zmienna celu	Nominalna

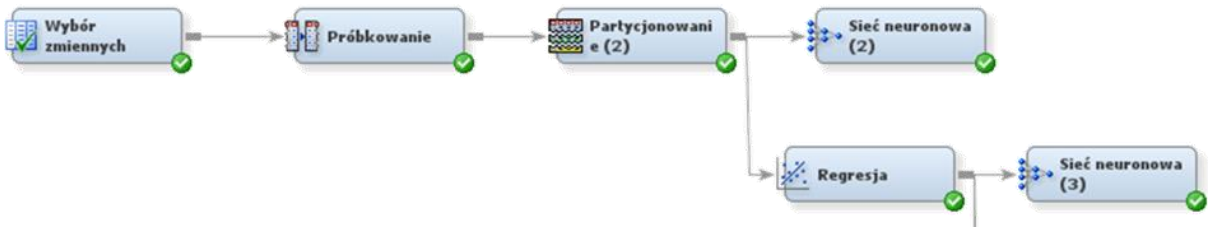
W węźle „Sieć neuronowa(2)" kryterium wyboru modelu będzie „Średni błąd". Rezultaty dla tego podejścia:

Tabela 9. Rezultat dla podejścia z węzłem "Wybór zmiennych"

Numer sieci	Odsetek błędnych klasyfikacji			Przeciętny błąd kwadratowy		
	Uczenie	Walidacja	Test	Uczenie	Walidacja	Test
2	0.418956	0.419122	0.416011	0.104025	0.10429	0.103981

Następnym podejściem do modelowania sieci neuronowej jest regresja logistyczna, która została użyta po węźle „Wybór zmiennych". Takie podejście było wymuszone ograniczeniami sprzętowymi i brakiem odpowiedzi programu na przydzielenie dodatkowej pamięci RAM.

Rysunek 15. Gałąź z węzłem "Regresja"



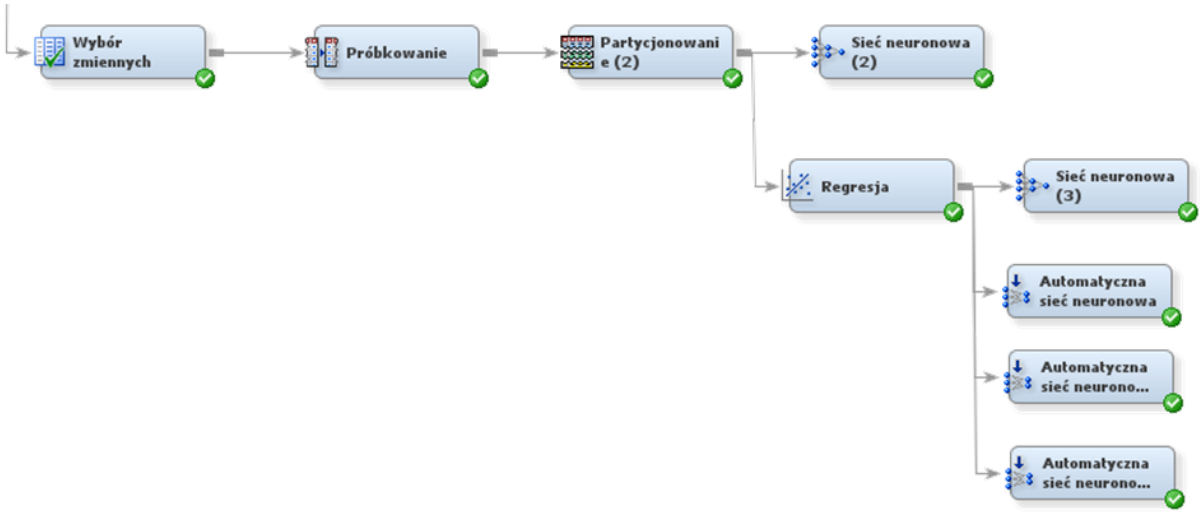
I tak rezultatem dla tego podejścia do modelowania sieci neuronowej jest:

Tabela 10. Rezultat dla podejścia z zastosowanie regresji logistycznej

Numer sieci	Odsetek błędnych klasyfikacji			Przeciętny błąd kwadratowy		
	Uczenie	Walidacja	Test	Uczenie	Walidacja	Test
3	0.418956	0.419122	0.416011	0.104025	0.10429	0.103981

Ostatnim podejściem do modelowania sieci neuronowej jest użycie węzła *„Automatyczna sieć neuronowa”*.

Rysunek 16. Gałąź z użyciem "Automatycznej sieci neuronowej"



W pierwszej i drugiej sieci neuronowej aktywowany został jedynie warunek „Tanh”, natomiast różnica pomiędzy tymi węzłami polega na różnej tolerancji, i tak pierwszy węzeł ma niską tolerancję, a drugi węzeł średnią. Natomiast trzeci węzeł ma niską tolerancję oraz więcej opcji aktywacji: bezpośrednia, normalna, sinus, tanh.

Tabela 11. Rezultat dla automatycznych sieci neuronowych

Numer sieci	Odsetek błędnych klasyfikacji			Przeciętny błąd kwadratowy		
	Uczenie	Walidacja	Test	Uczenie	Walidacja	Test

A1	0.402998	0.4041	0.401831	0.101264	0.101549	0.101187
A2	0.416586	0.419573	0.4144	0.104125	0.104268	0.103381
A3	0.793375	0.794404	0.794121	0.169699	0.169798	0.169681

Porównanie modeli:

Tabela 12. Tabela zbiorcza

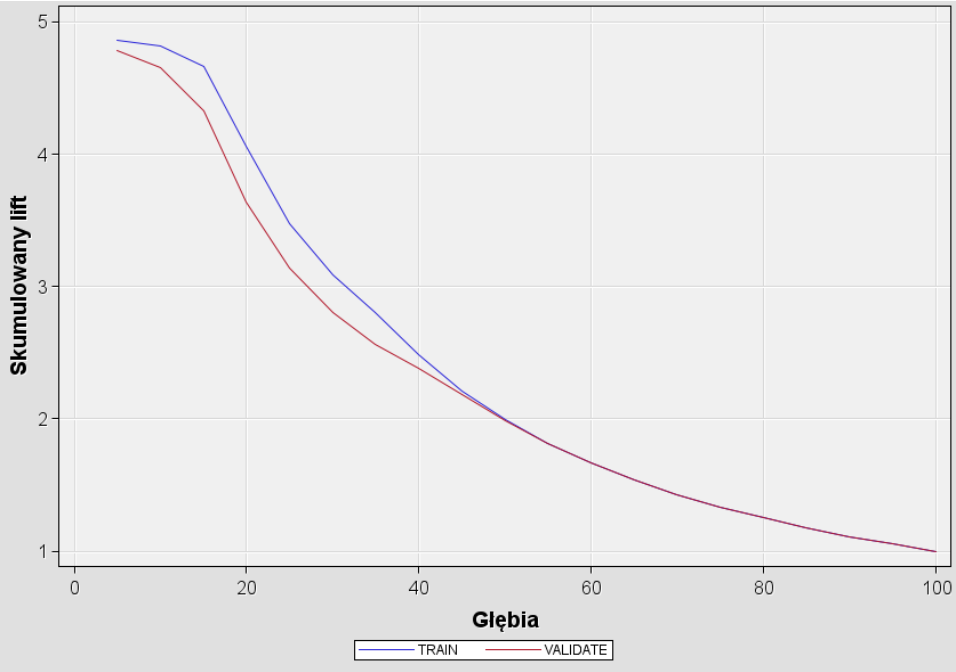
Numer sieci	Odsetek błędnych klasyfikacji			Przeciętny błąd kwadratowy		
	Uczenie	Walidacja	Test	Uczenie	Walidacja	Test
5	0.274514	0.36904	0.367679	0.069618	0.0887	0.087287
6	0.304273	0.386379	0.383091	0.080363	0.094716	0.093998
7	0.262096	0.264117	0.358432	0.065802	0.087877	0.086259
2	0.418956	0.419122	0.416011	0.104025	0.10429	0.103981
3	0.418956	0.419122	0.416011	0.104025	0.10429	0.103981
A1	0.402998	0.4041	0.401831	0.101264	0.101549	0.101187
A2	0.416586	0.419573	0.4144	0.104125	0.104268	0.103381
A3	0.793375	0.794404	0.794121	0.169699	0.169798	0.169681

Najmniejszy odsetek błędnych klasyfikacji oraz najmniejszy błąd kwadratowy otrzymujemy dla sieci neuronowej numer 7, czyli dla większej liczby ukrytych jednostek.

4.2 Analiza najlepszego przypadku

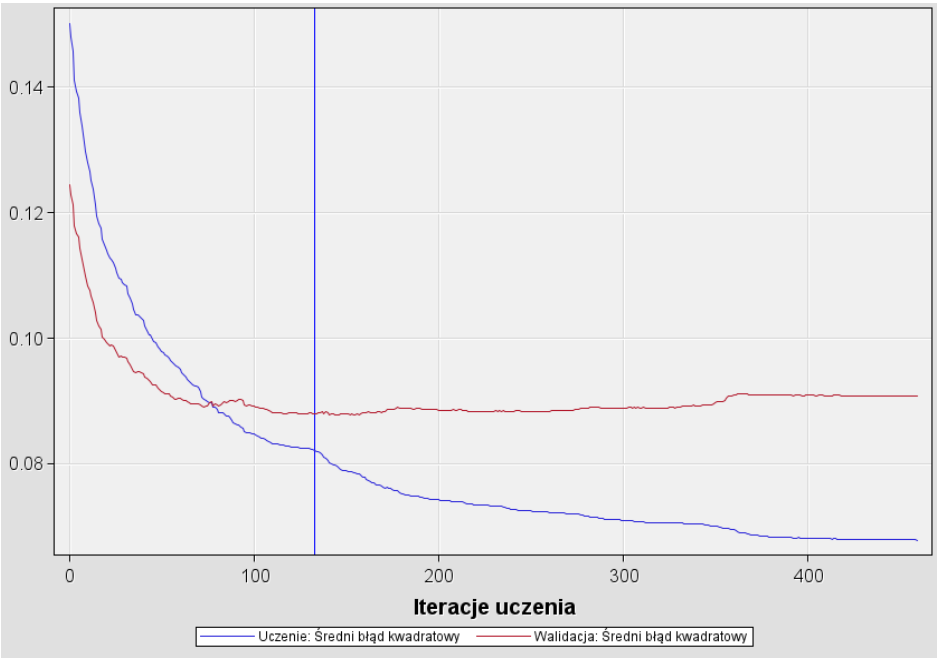
Tak jak wcześniej zostało zaznaczone najlepszy przypadek przy większej liczbie ukrytych jednostek, w tym przypadku 4.

Rysunek 17. Sieć neuronowa - Krzywa Lift



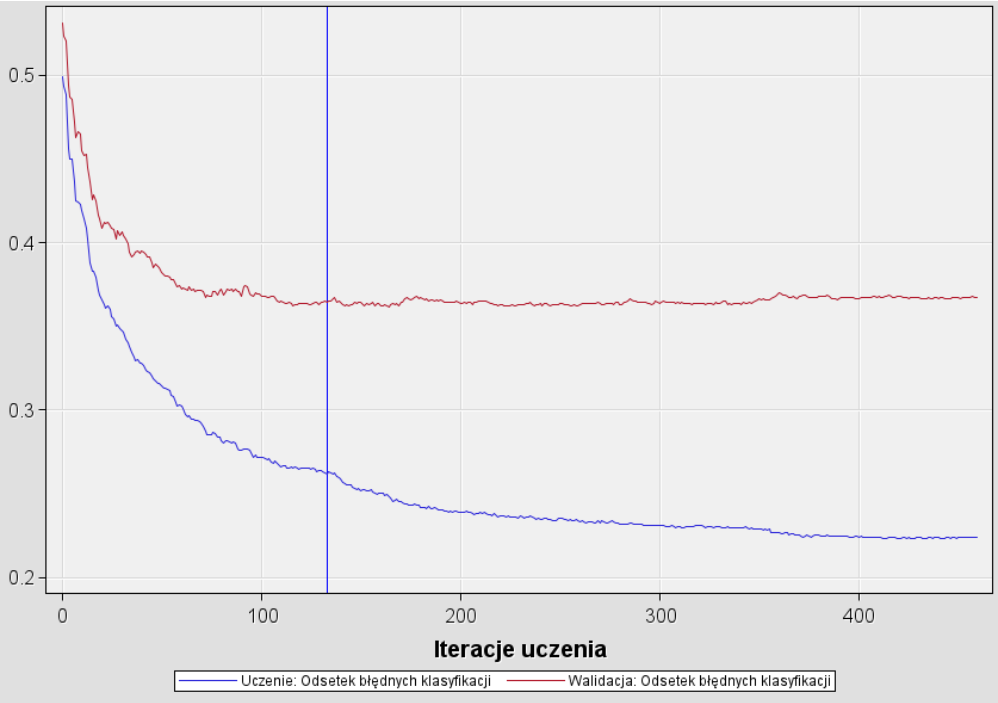
Krzywa Lift ukazuje, że model ma dobrą zbieżność dla krzywych zbioru uczącego i walidacyjnego oraz wysoką skuteczność poprawnej identyfikacji dla niskich głębokości, a wraz z zwiększaniem głębokości zbieżność wykresów polepsza się.

Rysunek 18. Sieć neuronowa - Błąd średniokwadratowy



Błąd średniokwadratowy maleje oraz utrzymuje się na niskim poziomie, tak więc statystyka ta mówi, że prognozy modelu są bliskie prawdziwym wartościom. Wraz z kolejnymi iteracjami uczenia błąd przestaje maleć i utrzymuje się mniej więcej na stałym poziomie dla zbioru walidacyjengo.

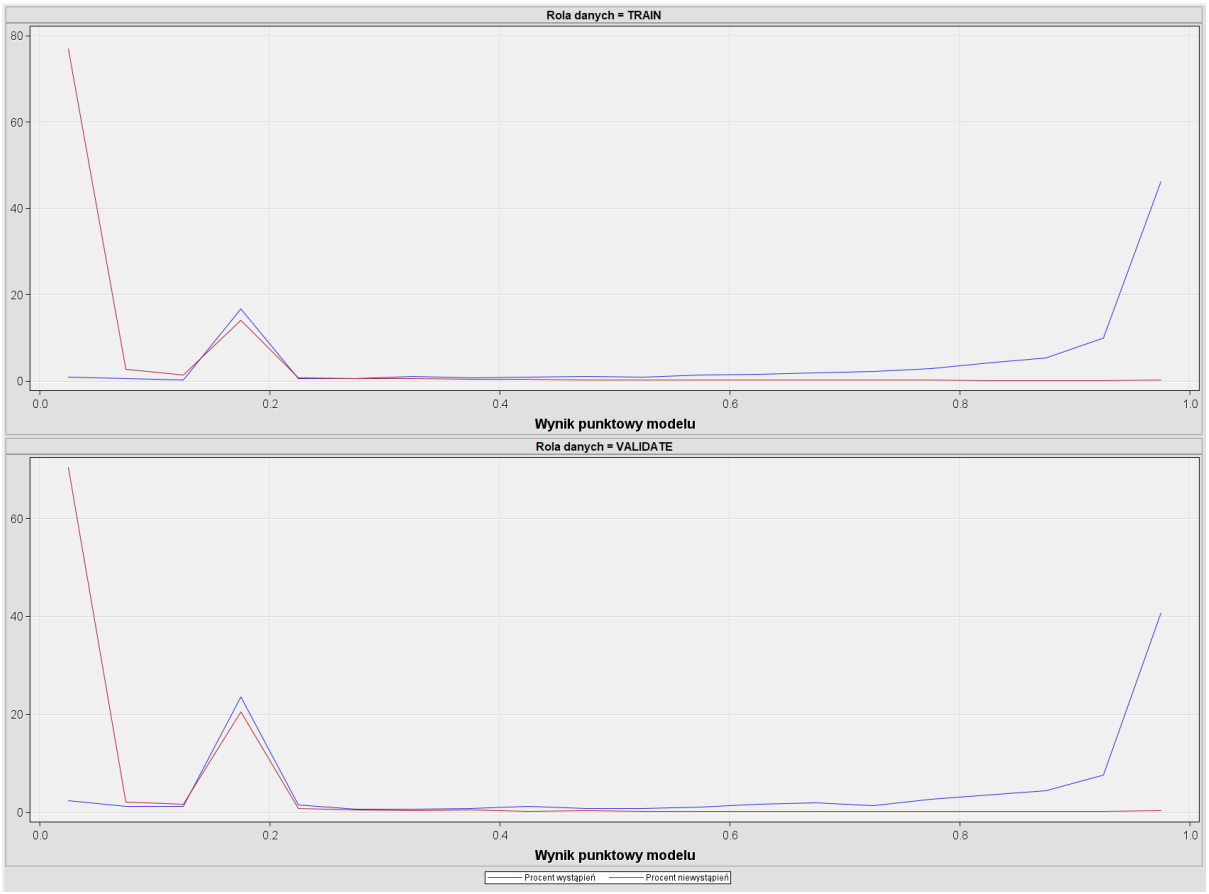
Rysunek 19. Sieć neuronowa - Odsetek błędnych klasyfikacji



Odsetek błędnych klasyfikacji maleje to pewnego momentu, natomiast po nim w przypadku zbioru walidacyjengo odsetek błędnych klasyfikacji utrzymuje się na stałym poziomie.

Odsetek błędnych klasyfikacji utrzymuje się w zbiorze testowym na poziomie 35,84%. Jest to wysoka wartość tzn. jeden na trzy przypadki zostanie błednie sklasyfikowany. Natomiast jest to najlepszy przypadek z możliwych do otrzymania dla tych danych wejściowych.

Rysunek 20. Sieć neuronowa - Wyniki punktowe modelu



Na obu wykresach widoczny jest szczyt dla wystąpień przy wyniku punktowym około 0.2. Sugeruje to, że model ma pewne trudności z jednoznacznym przypisaniem klasy w tej strefie, co może być spowodowane bliskością wartości z niedostatecznym rozróżnieniem cech w tej przestrzeni. Wysokie wartości punktowe (0.8-1.0) pokazują wyraźnie wyższy procent wystąpień, co wskazuje, że model poprawnie identyfikuje większość przypadków jako wystąpienia w tym zakresie. Świadczy to o dobrej predykcji modelu dla przypadków o wysokim prawdopodobieństwie wystąpienia.

Model sieci neuronowej może być przeuczony, ponieważ wraz z każdą kolejną iteracją nie osiągamy lepszych wartości dla krzywych błędu średniokwadratowego oraz odsetka błędnej klasyfikacji dla zbioru walidacyjnego, a w zbiorze uczenia zauważamy ciągły spadek. Ograniczenie liczby iteracji może pomóc, ale nie osiągniemy wtedy zbieżności wykresów, natomiast strata w poprawności predykcji wyników nie wzrasta wraz z kolejnymi iteracjami i utrzymuje się na podobnym poziomie. Jednym z sposobów może być zastosowanie jednej z metod regularyzacji.

5. Podsumowanie

Po wyestymowaniu i wyborze najlepszych modeli drzew decyzyjnych, lasów losowych oraz sieci neuronowych, przeanalizujemy te modele poprzez zestawienie ich ze sobą. Analiza ta obejmie kilka kluczowych aspektów, które pozwolą na dokładne porównanie wydajności i właściwości każdego z modeli.

Na początku skupimy się na przeanalizowaniu różnic i podobieństw w dokładności klasyfikacji wybranych modeli. Będziemy oceniać, jak dobrze każdy model radzi sobie z przewidywaniem wyników na zbiorach danych uczących i walidacyjnych. Dokładność klasyfikacji jest podstawowym wskaźnikiem efektywności modelu, dlatego zbadanie tych różnic pozwoli na wyciągnięcie wniosków dotyczących ogólnej skuteczności każdego modelu.

5.1 Dokładność klasyfikacji

Tabela 13. Dokładność klasyfikacji najlepszych modeli

Model	Dokładność klasyfikacji		
	Uczenie	Walidacja	Test
Sieć neuronowa 7	73,79%	73,59%	64,15%
Drzewo decyzyjne 3	78,53%	74,92%	75,08%
Las losowy	71,05%	70,63%	70,08%

Analizując wyniki przedstawione w tabeli 13. można zauważyć, że sieć neuronowa 7 osiągnęła wysoką dokładność na zbiorach uczącym i walidacyjnym, co sugeruje, że model dobrze radzi sobie z danymi, na których był trenowany. Jednak znaczący spadek dokładności na zbiorze testowym (64,15%) wskazuje na problem z generalizacją modelu na nowe dane, co może być symptomem przeuczenia (overfitting).

W przypadku drzewa decyzyjnego 3, wykazuje ono wysoką dokładność zarówno na zbiorze uczącym (78,53%) jak i na zbiorze walidacyjnym (74,92%). A co istotne, jego

dokładność na zbiorze testowym (75,08%) jest zbliżona do wyników na zbiorze walidacyjnym, co sugeruje, że ten model dobrze generalizuje i jest stabilny.

Las losowy osiągnął najniższą dokładność na zbiorze uczącym (71,05%) i walidacyjnym (70,63%). Są to wartości zbliżone do wyników osiągniętych na zbiorze testowym (70,08%). Niewielka różnica między dokładnością na tych zbiorach sugeruje, że model nie jest przeuczony.

Porównując te wyniki można zauważyć, że najlepsze właściwości pod względem stabilności i generalizacji możemy zaobserwować w modelu drzewa decyzyjnego 3. Wyniki na zbiorze testowym są spójne z wynikami na zbiorach uczącym i walidacyjnym, co sugeruje, że jest to najbardziej niezawodny model z analizowanych.

5.2 Krzywe ROC

Dodatkowo, przeprowadzimy analizę krzywych ROC (Receiver Operating Characteristic), które pozwolą ocenić zależność między czułością a specyficznością modeli. Analiza krzywych ROC dostarczy informacji o polu pod krzywą (AUC - Area Under the Curve), co jest miarą globalnej jakości modelu. Wyższe wartości AUC wskazują na lepszą zdolność modelu do rozróżniania klas.

Analizując dokładniej krzywe ROC wybranych modeli, zauważamy, że krzywa ROC dla modelu drzewa decyzyjnego przedstawia wysokie wartości AUC na wszystkich trzech zestawach danych: uczącym, walidacyjnym i testowym. Podobieństwo krzywych dla zestawów walidacyjnych i testowych potwierdza zdolność modelu do generalizacji. Model ten skutecznie klasyfikuje prawdziwe pozytywne przypadki przy minimalnej liczbie fałszywie pozytywnych klasyfikacji, co świadczy o jego wysokiej dokładności i efektywności w rozróżnianiu między klasami.

Krzywa ROC dla modelu lasów losowych ilustruje jego zdolność do rozróżniania pięciu klas cenowych samochodów. Wysoki AUC dla klas 0 i 1 świadczy o skutecznej identyfikacji tych klas z minimalnym błędem. Niższe wartości AUC dla klas 2, 3 i 4 sugerują pewne trudności w dokładnym rozróżnianiu tych klas, co może być spowodowane podobieństwami cech między nimi lub niewystarczającą reprezentacją tych klas w danych treningowych.

Oba modele wykazują różne zalety i ograniczenia. Model drzewa decyzyjnego odznacza się wysoką ogólną dokładnością i zdolnością do generalizacji na różnych zestawach danych. Z kolei model lasów losowych, pomimo zdolności do skutecznego rozróżniania niektórych klas, może wymagać dalszej optymalizacji, zwłaszcza dla mniej licznie reprezentowanych klas.

Wnioski

Podsumowując, po dokładnej analizie wybranych modeli oraz kierując się własną intuicją, zdecydowałabym się na model drzewa decyzyjnego ze względu na jego wysokie parametry oraz skuteczność w klasyfikacji. Model ten wykazał się doskonałą zdolnością do generalizacji na różnych zestawach danych, co potwierdzają wysokie wartości AUC i stabilność krzywych ROC na zbiorach uczącym, walidacyjnym i testowym.

Jednakże należy zauważyć, że pozostałe modele również prezentują zadowalające wyniki, co czyni je godnymi uwagi do dalszej analizy czynników wpływających na cenę samochodów. Modele te mogą wnieść dodatkowe perspektywy i wnioski, zwłaszcza w kontekście różnorodnych cech i ich wpływu na predykcję cen.

Przeprowadzona analiza pozwoliła na zidentyfikowanie kluczowych czynników, które wpływają na cenę samochodu. Pierwszym z tych czynników jest przebieg samochodu w kilometrach - obrazowany przez zmienną odometer. Im mniejszy przebieg tym samochód jest bardziej wydajny i mniej narażony na wszelkie awarie. Tak więc mniejsza ilość kilometrów przebiegu często kojarzona z lepszą jakością a co za tym idzie wydajnością samochodu, co może prowadzić do wyższej ceny.

Kolejnym istotnym czynnikiem jest zmienna seller, czyli nazwa sprzedawcy samochodu. Z analizy modeli wynika, że ma ona bezpośredni wpływ na cenę pojazdu. Sprzedawcy o bardziej znanych i zaufanych markach mogą żądać wyższych cen, ponieważ ich reputacja często wiąże się z lepszą jakością usług i większą pewnością transakcji.

Sam model samochodu ma znaczący wpływ na jego cenę, ponieważ różne modele są projektowane z myślą o różnych segmentach rynku i oferują różne poziomy wyposażenia, technologii i prestiżu. Samochody renomowanych marek oraz modele z zaawansowanymi

specyfikacjami technicznymi, luksusowym wyposażeniem i nowoczesnymi systemami bezpieczeństwa są zazwyczaj wyceniane wyżej. Ponadto, popularność i popyt na konkretny model, a także jego rok produkcji i efektywność paliwowa, również przyczyniają się do zróżnicowania cen.

Ostatnią znaczącą zmienną jest condition, czyli ocena stanu technicznego pojazdu. Stan techniczny wpływa na wartość samochodu, ponieważ odzwierciedla jego niezawodność, bezpieczeństwo i koszty eksploatacji. Pojazdy w doskonałym stanie z pełną dokumentacją serwisową są wyżej wyceniane, podczas gdy te z uszkodzeniami czy brakami w historii serwisowej są tańsze.

Wszystkie te czynniki mają bezpośredni wpływ na jakość samochodu, komfort jego użytkowania i zadowolenie kierowcy. Niski przebieg samochodu, znana marka, popularność modelu oraz jego stan techniczny to cechy, na które zwracają uwagę kupujący. Świadczą one o ich jakości i niezawodności. Dlatego te czynniki mają istotne znaczenie w sprzedaży samochodów.

Spis Tabel

Spis Wykresów

Kody

Wstępna obróbka danych

```
import pandas as pd

# Wczytywanie danych z pliku CSV
file_path = 'C:/Users/48514/Downloads/car_prices_3.csv/car_prices_3.csv'
data = pd.read_csv(file_path)

# Sprawdzenie, czy są jakiekolwiek puste wartości w DataFrame
null_values = data.isnull().values.any()

if null_values:
    print("Istnieją puste wartości w danych.")
else:
    print("Nie ma pustych wartości w danych.")

# Liczba brakujących wartości w każdej kolumnie
missing_values_per_column = data.isnull().sum()
print("Liczba brakujących wartości w każdej kolumnie:")
print(missing_values_per_column)

# Liczba obserwacji z przynajmniej jedną brakującą wartością
observations_with_missing_values = data.isnull().any(axis=1).sum()
print(f"\nLiczba obserwacji z przynajmniej jedną brakującą wartością:
{observations_with_missing_values}")

# Liczba obserwacji bez brakujących wartości
observations_without_missing_values = data.notnull().all(axis=1).sum()
print(f"Liczba obserwacji bez brakujących wartości:
{observations_without_missing_values}")

# Całkowita liczba brakujących wartości w DataFrame
total_missing_values = data.isnull().sum().sum()
print(f"Całkowita liczba brakujących wartości w DataFrame: {total_missing_values}")

# Usunięcie wierszy z brakującymi wartościami
data_cleaned = data.dropna()

# Definiowanie kwintylów
data_cleaned.loc[:, 'price_category'] = pd.qcut(data_cleaned['sellingprice'], q=5,
labels=['Bardzo tanie', 'Tanie', 'Średnie', 'Drogie', 'Bardzo drogie'])
```

```
# Wyświetlenie kilku pierwszych wierszy
print(data_cleaned[['sellingprice', 'price_category']].head())

# Zapisanie oczyszczonego zestawu danych do nowego pliku CSV
cleaned_file_path = 'C:/Users/48514/Downloads/car_prices_3.csv/car_prices_3_cleaned.csv'
data_cleaned.to_csv(cleaned_file_path, index=False)

print(data_cleaned)
```

Lasy Losowe

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

data = pd.read_csv(car_prices_3_cleaned.csv)

data_cars = data.drop('mmr', axis=1)
data_cars.head()

sample_size = int(len(data_cars) * 0.10)
sample = data_cars.sample(n=sample_size, replace=False)
print(sample)

X = pd.get_dummies(sample.drop(['price_category'], axis=1))
y = sample['price_category']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

las = RandomForestClassifier(n_estimators=100, random_state=1)
las.fit(X_train, y_train)

from sklearn.metrics import accuracy_score

y_pred = las.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('Dokładność predykcji:', accuracy)

scores_ = cross_val_score(las, X, y, cv=5)

print('Dokładność predykcji:', scores_)
```

```
print('Średnia dokładność:', scores_.mean())

from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc
import numpy as np

# Binaryzacja etykiet
y = label_binarize(y_test, classes=np.unique(y_test))
n_classes = y.shape[1]

# Predykcja prawdopodobieństw dla wszystkich klas
y_score = las.predict_proba(X_test)

# Obliczenie krzywej ROC i AUC dla każdej klasy
fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Wyświetlenie krzywej ROC dla każdej klasy
plt.figure()
for i in range(n_classes):
    plt.plot(fpr[i], tpr[i], label='Klasa %d (obszar = %0.2f)' % (i, roc_auc[i]))

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Krzywa ROC dla poszczególnych klas')
plt.legend(loc="lower right")
plt.show()
```