Analiza zależności cen wybranych samochodów osobowych od ich parametrów i danych technicznych

Michał Maksanty

Maj 2024

Spis treści

1 Wstęp

Poniższa praca będzie zajmować się badaniem zależności cen wybranych modeli samochodów osobowych (Volkswagen Golf, BMW Seria 3, Opel Corsa - wybrane generacje) w zależności od:

- pojemności silnika,
- roku produkcji,
- mocy,
- rodzaju paliwa,
- typu nadwozia,
- przebiegu,
- koloru,
- stanu,
- skrzyni biegów,
- pochodzenia.

Następnie zostaną utworzone modele regresji i na podstawie przyjętej w późniejszej części pracy metryki, zostanie wybrany najlepszy model.

Pomysł projektu zainspirowany jest faktem dostępności dużej ilości samochodów na rynku wtórnym. Z tej racji chciałem sprawdzić jakie zależności rzeczywiście mają zastosowanie w przypadku takich danych - od czego cena jest bardziej zależna, a od czego mniej - taka analiza może ułatwić ewentualne poszukiwania samochodu i potencjalnie zwiększyć szansę na znalezienie auta w niższej cenie. A jej ciekawym efektem będzie oczywiście wybrany model najlepiej opisujący analizowane dane.

2 Zbiór danych

2.1 Źródła danych

Wstępne założenia projektu zakładały wykorzystanie jako źródła danych o samochodach platform internetowych: OLX oraz OTOMOTO.

2.2 Pobieranie danych

Dane z serwisów pozyskałem za pomocą scrapingu. Wykorzystałem do tego celu popularne biblioteki języka Python: *requests*, *BeautifulSoup* oraz *csv*.

Program, po podaniu linku z ofertami samochodów, analizuje, przetwarza i pobiera dane ze wszystkich ofert samochodów dostępnych pod podanym linkiem oraz na kolejnych stronach, jeżeli treść jest podzielona metodą paginacji.

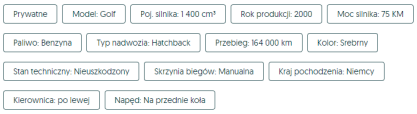
Po zebraniu danych z obu serwisów i przeprowadzeniu ich krótkiej analizy, łatwo można było zauważyć, że dane z OTOMOTO były w znacznej większości kompletne, natomiast te z OLX zawierały sporo wartości pustych. Wynika to z

faktu, że strona OTOMOTO z ofertą auta, ma przeznaczone pola obligatoryjne na podanie różnych parametrów samochodu, dzięki czemu, poza nielicznymi przypadkami, wszystkie dane były kompletne. Natomiast w przypadku OLX, wybór dostępnych na stronie informacji o samochodzie zależy całkowicie od sprzedającego, przez co puste pola występowały bardzo często. Z tego względu oraz faktu, że OTOMOTO posiada bardzo dużą bazę dostępnych samochodów, w pełni wystarczającą do przeprowadzenia analizy na rzecz tego projektu, na tym etapie zdecydowałem się zrezygnować z danych z OLX i skupić się jedynie na części z OTOMOTO.

	brand	model	generation	eng_cap	prod_year	power	fuel_type	car_body	mileage	color	condition	transmission	origin	price	source
2	Opel	Corsa	Unknown	1 000 cm³	2004	60 KM	Benzyna	Hatchback	288 000 km	Inny kolor	Nieuszkodzony	Manualna	Unknown	4 999 zł	OLX
3	Opel	Corsa	Unknown	1 200 cm³	2011	80 KM	Benzyna	Sedan	156 000 km	Niebieski	Nieuszkodzony	Manualna	Polska	9 800 zł	OLX
4	Opel	Corsa	D (2006-2014)	1 248 cm3	2012	95 KM	Diesel	Auta miejskie	149 000 km	Czarny	Używane	Manualna	Niemcy	15 500PLN	OTOMOTO
5	Opel	Corsa	D (2006-2014)	1 229 cm3	2013	85 KM	Benzyna	Auta małe	151 093 km	Szary	Używane	Manualna	Niemcy	20 900PLN	OTOMOTO
6	Opel	Corsa	D (2006-2014)	1 398 cm3	2014	87 KM	Benzyna	Auta miejskie	142 935 km	Biały	Używane	Manualna	Niemcy	21 600PLN	OTOMOTO
7	Opel	Corsa	D (2006-2014)	1 229 cm3	2013	85 KM	Benzyna	Kompakt	40 227 km	Czarny	Używane	Manualna	Polska	25 500PLN	OTOMOTO
8	Opel	Corsa	F (2019-)	1 199 cm3	2020	100 KM	Benzyna	Auta miejskie	33 000 km	Czarny	Używane	Automatyczna	Unknown	62 900PLN	OTOMOTO
9	Opel	Corsa	E (2014-2019)	1 398 cm3	2018	90 KM	Benzyna+LPG	Kompakt	193 000 km	Biały	Używane	Manualna	Unknown	33 210PLN	OTOMOTO
10	Opel	Corsa	D (2006-2014)	1 229 cm3	2007	80 KM	Benzyna	Auta małe	123 484 km	Srebrny	Używane	Manualna	Unknown	8 999PLN	OTOMOTO
11	Opel	Corsa	D (2006-2014)	1 398 cm3	2013	87 KM	Benzyna	Auta miejskie	174 727 km	Czarny	Używane	Manualna	Niemcy	23 900PLN	OTOMOTO
12	Opel	Corsa	D	998 cm³	2009	65 KM	Benzyna	Hatchback	175 558 km	Niebieski	Nieuszkodzony	Manualna	Inny	8 000 zł do negocjacji	OLX
13	Opel	Corsa	D (2006-2014)	1 229 cm3	2007	80 KM	Benzyna	Coupe	126 800 km	Błękitny	Używane	Manualna	Unknown	9 900PLN	OTOMOTO
14	Opel	Corsa	E (2014-2019)	1 398 cm3	2016	75 KM	Benzyna	Kompakt	74 000 km	Szary	Używane	Manualna	Polska	37 500PLN	OTOMOTO
15	Opel	Corsa	1.7	1 700 cm³	2008	125 KM	Diesel	Coupe	224 000 km	Srebrny	Nieuszkodzony	Manualna	Unknown	8 900 zł	OLX

Rysunek 1: Reprezentacyjny zbiór danych z OTOMOTO i OLX, pokazujący częste wartości puste dla danych z OLX

Rysunek 2: Dane obligatoryjne na OTOMOTO



Rysunek 3: Dane opcjonalne na OLX

2.3 Oczyszczanie danych

Przed właściwą analizą danych, wymagają one wstępnego przetworzenia i oczyszczenia.

```
<class 'pandas.core.frame.DataFrame'>
Index: 6867 entries, 1550 to 860
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   brand         6867 non-null   object
 1   model         6867 non-null   object
 2   generation    6805 non-null   object
 3   eng_cap       6782 non-null   object
 4   prod_year     6867 non-null   int64
 5   power         6861 non-null   object
 6   fuel_type     6867 non-null   object
 7   car_body      6867 non-null   object
 8   mileage       6867 non-null   object
 9   color         6867 non-null   object
 10  condition     6867 non-null   object
 11  transmission  6865 non-null   object
 12  origin        4154 non-null   object
 13  price         6867 non-null   object
 14  source        6867 non-null   object
dtypes: int64(1), object(14)
memory usage: 858.4+ KB
```

Rysunek 4: Typy danych do analizy

	brand	model	generation	eng_cap	prod_year	power	fuel_type	car_body	mileage	color	condition	transmission	origin	price	source
1550	Volkswagen	Golf	VII (2012-2020)	1 984 cm3	2017	310 KM	Benzyna	Kompakt	77 000 km	Biały	Używane	Automatyczna	NaN	114 000PLN	OTOMOTO
1221	Volkswagen	Golf	VIII (2020-)	1 968 cm3	2022	150 KM	Diesel	Kompakt	32 000 km	Szary	Używane	Automatyczna	Polska	95 000PLN	OTOMOTO
1941	Volkswagen	Golf	VIII (2020-)	1 968 cm3	2020	150 KM	Diesel	Kompakt	76 020 km	Czarny	Używane	Automatyczna	Francja	86 900PLN	OTOMOTO
573	BMW	Seria 3	E90/E91/E92/E93 (2005-2012)	2 996 cm3	2005	258 KM	Benzyna	Sedan	288 858 km	Szary	Używane	Manualna	NaN	32 500PLN	OTOMOTO
173	Opel	Corsa	D (2006-2014)	1 229 cm3	2009	80 KM	Benzyna	Kompakt	152 000 km	Czarny	Używane	Manualna	NaN	18 500PLN	OTOMOTO

Rysunek 5: Reprezentacyjny zbiór danych z nieusuniętymi jednostkami

Jak widać na powyższych spisach - jedynie rok produkcji jest od samego początku zapisany jako poprawny typ danych. Aby sprowadzić resztę danych do właściwej formy należy na tym etapie dla wszystkich wartości liczbowych usunąć jednostkę, by móc następnie rzutować wartości na typy liczbowe.

Dodatkowo dla ceny należy zamienić wartości podane w walutach zagranicznych na PLN (wykorzystana zostanie do tego biblioteka forex-python [3])

	brand	model	generation	eng_cap	prod_year	power	fuel_type	car_body	mileage	color	condition	transmission	origin	price	source
1550	Volkswagen	Golf	VII (2012-2020)	1984.0	2017	310.0	Benzyna	Kompakt	77000	Biały	Używane	Automatyczna	NaN	114000.0	OTOMOTO
1221	Volkswagen	Golf	VIII (2020-)	1968.0	2022	150.0	Diesel	Kompakt	32000	Szary	Używane	Automatyczna	Polska	95000.0	OTOMOTO
1941	Volkswagen	Golf	VIII (2020-)	1968.0	2020	150.0	Diesel	Kompakt	76020	Czarny	Używane	Automatyczna	Francja	86900.0	OTOMOTO
573	BMW	Seria 3	E90/E91/E92/E93 (2005-2012)	2996.0	2005	258.0	Benzyna	Sedan	288858	Szary	Używane	Manualna	NaN	32500.0	OTOMOTO
173	Opel	Corsa	D (2006-2014)	1229.0	2009	80.0	Benzyna	Kompakt	152000	Czarny	Używane	Manualna	NaN	18500.0	OTOMOTO

Rysunek 6: Reprezentacyjny wycinek danych po rzutowaniu typów

3 Analiza i dalsze przetwarzanie przygotowanych danych

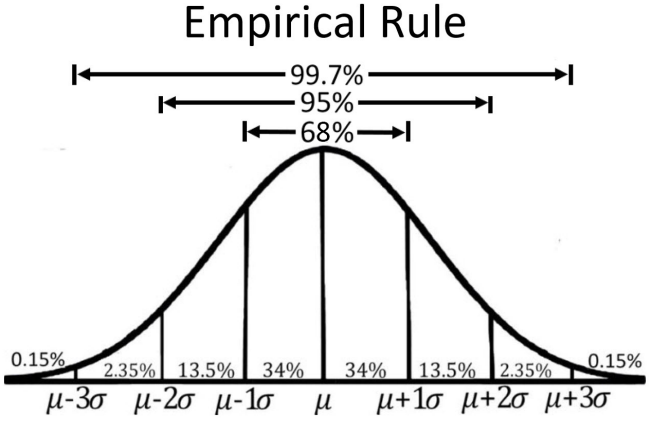
3.1 Eksploracyjna analiza danych

Do celów wstępnej analizy wizualnej danych zostało wykorzystane narzędzie ProfileReport z modułu pythona: ydata_profiling.

3.2 Funkcja do analizy wartości odstających (outlierów)

Zanim poszczególne zmienne zostaną dogłębniej przeanalizowane, warto przygotować funkcję, która pomoże nam odfiltrować nadzwyczajnie, odstającą od reszty część danych - zwaną *outlierami*. Zwykle nie chcemy analizować takich danych, gdyż w większości przypadków są to informacje nieprawdziwe, powstałe na skutek różnego rodzaju błędów i mogłyby one negatywnie wpływać na wyniki naszego modelu.

Utworzona funkcja klasyfikuje wartości jako odstające na podstawie ich **zscore** - jest to współczynnik oznaczający o ile odchyleń standardowych dana wartość różni się od oczekiwanej. Jeżeli wartość bezwzględna z **zscore** będzie większa od przyjętego progu, wówczas próbka zostanie sklasyfikowana jako odstająca. Za nasz próg przyjmiemy liczbę 3 - gdyż zgodnie z *regułą trzech sigm* [10]: dla danych o rozkładzie normalnym, 99,7% próbek, będzie mieścić się w granicach trzech odchyleń standardowych, po obu stronach wartości oczekiwanej.



Rysunek 7: Reguła trzech sigm

3.3 Usuwanie duplikatów

Pomimo, że teoretycznie na stronie internetowej, z ofertami samochodowymi nie powinno być żadnych powtórek, to jednak należy wziąć pod uwagę, że pobranie

tak dużej ilości danych zajmuje trochę czasu (ponad 4 godziny dla prawie 7000 samochodów) i przez ten czas niektóre oferty mogą zostać dodane, usunięte lub odświeżone i pomimo ich pobrania, mogą one pojawić się znowu na następnych kartach. Dlatego też warto na tym etapie pozbyć się spowodowanych w ten sposób powtórzeń.

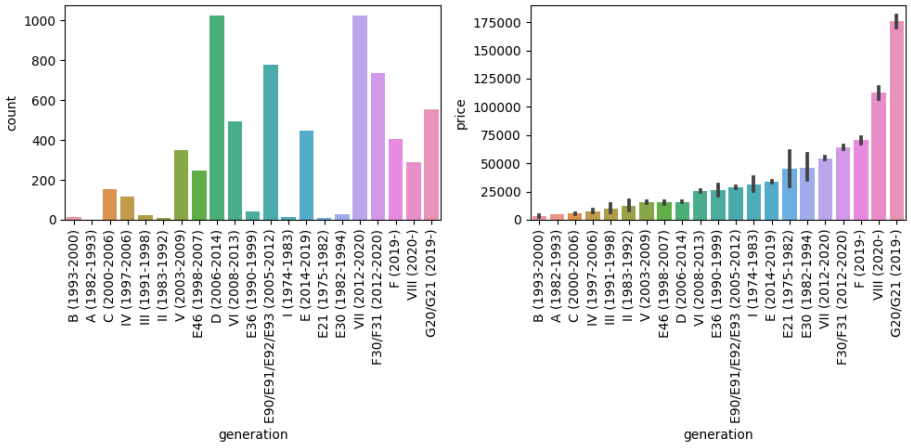
3.4 Źródło

Zmienna decyzyjna źródło, jest artefaktem pozostałym, po wersji scrapera, który pobierał dane również z OLX. Teraz, kiedy źródłem danych pozostało OTOMOTO, w tym polu występuje tylko jedna możliwa wartość. Można je więc od razu usunąć z naszych rozważań.

3.5 Marka, model

Zmienne decyzyjne marka, model oraz generacja, wspólnie tworzą 3-elementową krotkę, jednoznacznie wyróżniającą dany rodzaj samochodu na tle innych (np. Volkswagen Golf V). Jednak warto zwrócić uwagę, że (w przypadku wybranych do analizy pojazdów), sama generacja będzie wystarczająca do jednoznacznej identyfikacji typu pojazdu - gdyż konkretna generacja ma unikatową nazwę w obrębie danego modelu, który to przynależy pod daną markę - i nazwa tej generacji nie powtarza się dla żadnego innego modelu, jakiejkolwiek innej marki. Dzięki takiej zależności możemy zmniejszyć złożoność naszych danych i usunąc z nich obie te zmienne, zostawiając tylko generację.

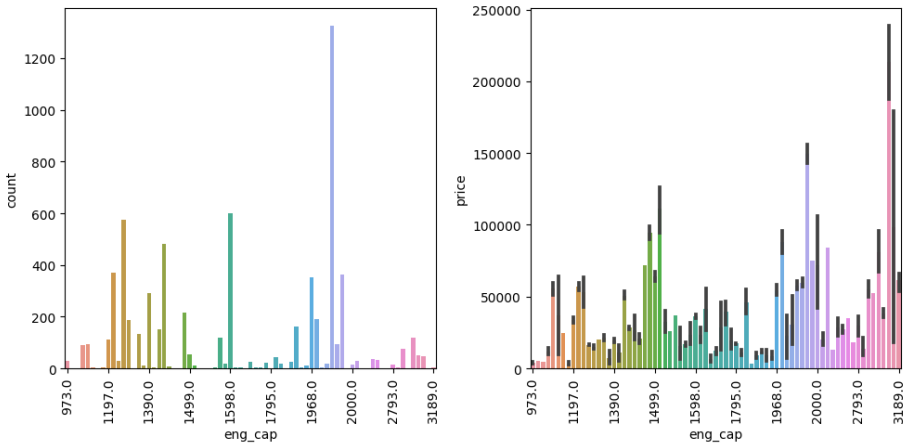
3.6 Generacja



Rysunek 8: Histogram generacji oraz wykres zależności ceny od generacji

Jak można zauważyć z powyższych wykresów, występuje widoczna zależność pomiędzy generacją samochodu a jego ceną. Histogram również nie daje nam powodów do usunięcia tej zmiennej decyzyjnej z analizy. Poza tym, usuwając ją, stracilibyśmy możliwość sprawdzenia ceny samochodu, dla konkretnego typu pojazdu (jako zmienne losowe, zostałyby tylko parametry auta), a przecież takie było założenie naszego projektu - by estymować cenę **konkretnego auta** na podstawie jego parametrów technicznych - zatem generacja samochodu zdecydowanie musi pozostać w zbiorze analizowanych danych.

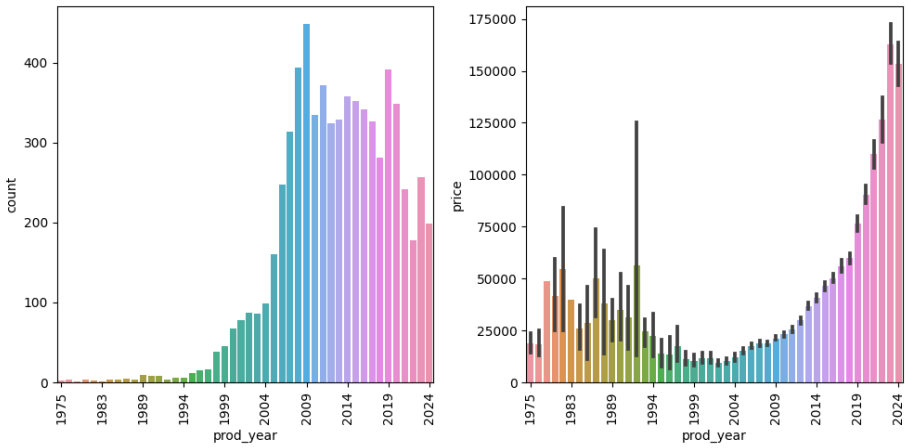
3.7 Pojemność silnika



Rysunek 9: Histogram pojemności silnika oraz wykres zależności ceny od pojemności silnika

Jak widać na wykresie, nie mamy tutaj do czynienia z silną korelacją pomiędzy ceną a pojemnością silnika. Wykres rzeczywiście ma tendencję wzrostową, jednakże jest ona zaszumiona i mocno się waha. Z tego względu, po utworzeniu modelu zrobiłem eksperyment, w którym testowałem wpływ usunięcia pojemności silnika ze zbioru danych na metryki modeli. Okazało się, że wyniki były nieco lepsze przy pozostawieniu tej części danych - więc tak uczynię.

3.8 Rok produkcji



Rysunek 10: Histogram lat produkcji i wykres zależności ceny od roku produkcji

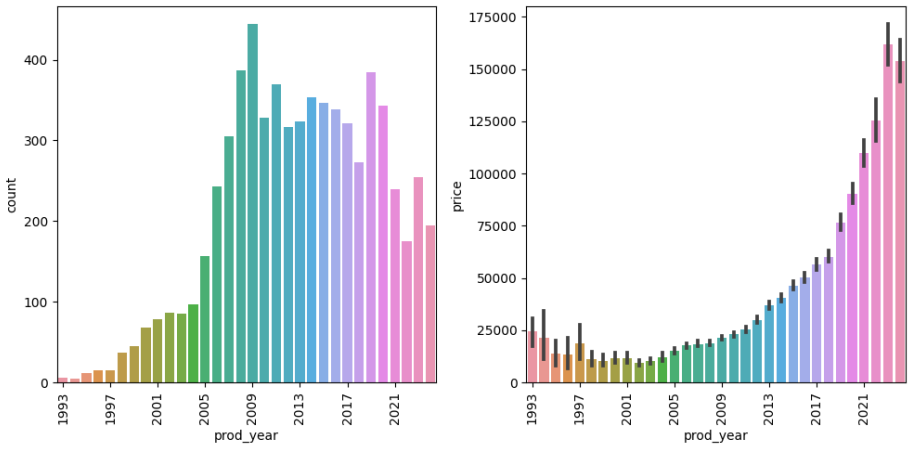
Jak można odczytać z wykresu, rok produkcji jest silnie skorelowany z ceną. Wyłania nam się ciekawa, paraboliczna zależność - najnowsze samochody są najdroższe, im auto jest starsze, tym cena jest niższa, ale do czasu. W pewnym momencie (około 2000 roku) wykres odbija w drugą stronę - samochody z ubiegłego wieku są uznawane za klasyki przez co ich cena (wraz z wiekiem) zaczyna rosnąć.

Z histogramu możemy zobaczyć, że wykres rozpoczyna się od 1975 roku, pomimo bardzo małej ilości samochodów aż do 1990 roku. Z tego względu warto odfiltrować dane pod kątem wartości odstających. Użyjemy do tego celu wcześniej wprowadzonej funkcji 3.2.

```
Outliers of prod_year: 58
```

	generation	eng_cap	prod_year	power	fuel_type	car_body	mileage	color	condition	transmission	origin	price
561	I (1974-1983)	1085.0	1981	50.0	Benzyna	Kompakt	306000	Czerwony	Używane	Manualna	NaN	24900.0
237	E30 (1982-1994)	1585.0	1989	99.0	Benzyna	Coupe	204000	Szary	Używane	Manualna	NaN	36999.0
1034	E30 (1982-1994)	2793.0	1989	193.0	Benzyna	Coupe	200000	Pomarańczowy	Używane	Manualna	NaN	59900.0
2168	I (1974-1983)	1457.0	1977	70.0	Benzyna	Auta małe	183855	Czerwony	Używane	Manualna	NaN	17700.0
1609	E30 (1982-1994)	1596.0	1990	100.0	Benzyna+LPG	Sedan	384000	Czerwony	Używane	Manualna	NaN	33600.0

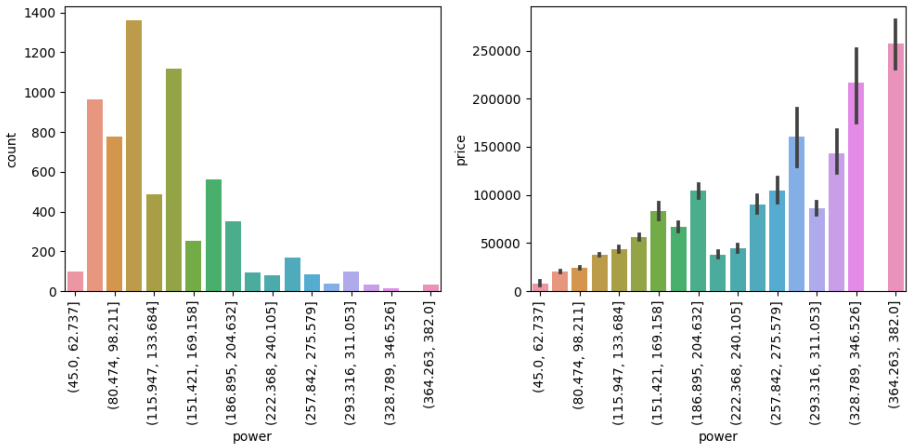
Rysunek 11: Reprezentacyjne wartości odstające roku produkcji



Rysunek 12: Histogram lat produkcji i wykres zależności ceny od roku produkcji po odfiltrowaniu wartości odstających

Jak widać na nowym wykresie, nasza parabola przyjęła bardziej gładki kształt, a rozłożenie wartości na histogramie bardziej przypomina rozkład normalny niż wcześniej.

3.9 Moc

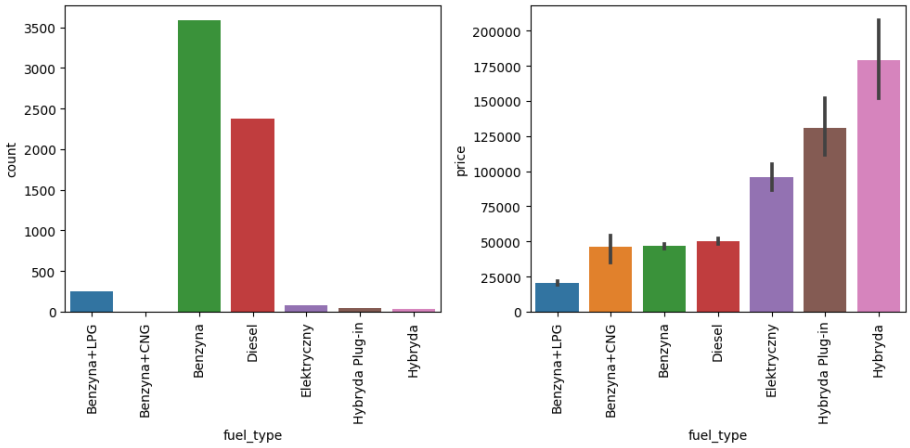


Rysunek 13: Histogram mocy i wykres zależności ceny od mocy

Wykres przedstawia całkiem gładką, wzrostową charakterystykę mocy w stosunku do ceny samochodu. Po odfiltrowaniu danych pod kątem ewentualnych wartości odstających - moc zostanie wykorzystana przy późniejszym uczeniu

modelu.

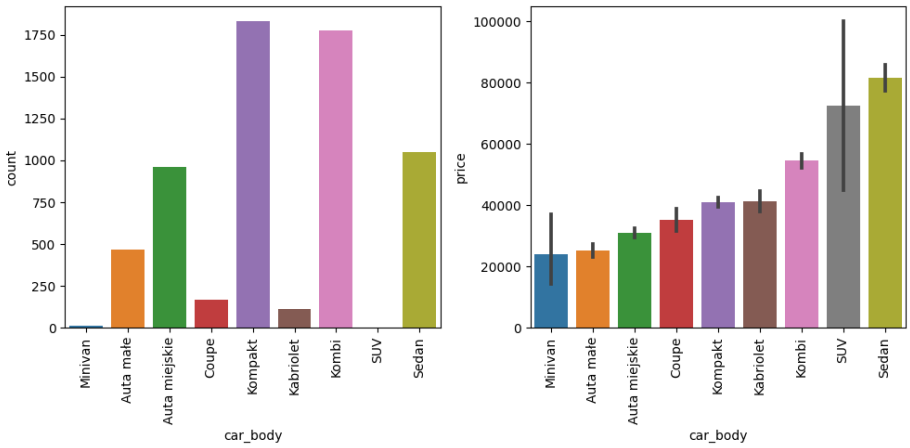
3.10 Rodzaj paliwa



Rysunek 14: Histogram rodzaju paliwa i wykres zależności ceny od rodzaju paliwa

Jak widać na histogramie - oprócz samochodów z silnikiem diesla i benzynowym, nie mamy zbyt wiele danych na temat innych rodzajów paliwa - z tego względu, pomimo zróżnicowania poziomu cen w zależności od użytego paliwa, nie będziemy rozważać tych danych w przypadku naszego modelu - dane są zbyt nierównomiernie rozłożone.

3.11 Typ nadwozia



Rysunek 15: Histogram typu nadwozia oraz wykres zależności ceny od nadwozia samochodu

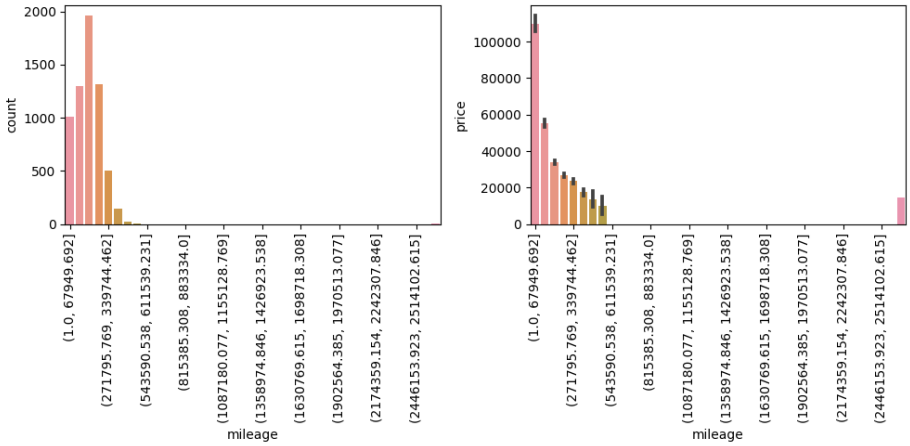
Pomimo, że widać znaczące różnice w cenie dla samochodów z różnymi typami nadwozia, podobnie jak z rodzajami paliwa, nie mamy zbyt wiele próbek danych dla części z tych typów.

Typ nadwozia	**Liczba**	**% wszystkich**
Kompakt	1829	28.681198
Kombi	1777	27.865768
Sedan	1050	16.465423
Auta miejskie	961	15.069782
Auta małe	465	7.291830
Coupe	167	2.618786
Kabriolet	115	1.803356
Minivan	11	0.172495
SUV	2	0.031363

Tabela 1: Zestawienie samochodów z podziałem na różne typy nadwozi

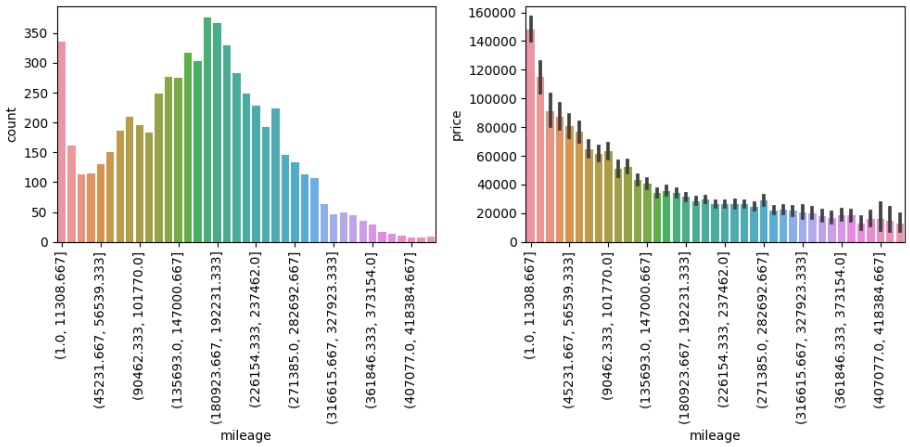
Jak widać w powyższej tabeli, SUV'y i minivany nie stanowią nawet 1% wszystkich danych i tylko 4 ze wszystkich rodzajów nadwozi przekraczają 10% udziału wśród danych. Ze względu na takie zróżnicowanie, nie będziemy rozważać dalej tej zmiennej decyzyjnej.

3.12 Przebieg



Rysunek 16: Histogram przebiegu oraz wykres zależności ceny od przebiegu samochodu

Jak widać na obu diagramach, większość danych umiejscowiona jest w przedziale do około 400 000km. Poza tym przedziałem na wykresie widać tylko jeden słupek na wartości ponad 2 500 000km. Nie widzę potrzeby brania pod uwagę całego tego przedziału tylko dla pojedynczych dodatkowych wartości, zatem przefiltruję je pod kątem wartości odstających.

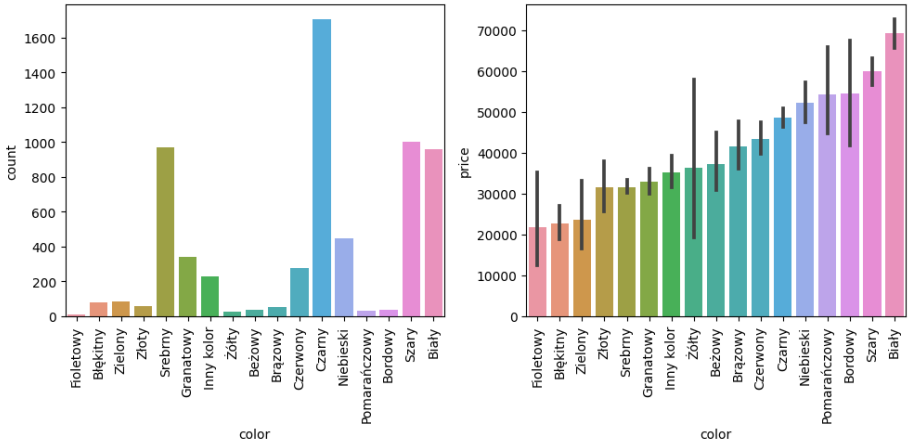


Rysunek 17: Histogram przebiegu oraz wykres zależności ceny od przebiegu samochodu po usunięciu wartości odstających

Teraz histogram przebiegu bardzo przypomina rozkład Gaussowski. Zależ-

ność ceny od przebiegu również się wyklarowała. Korelacja okazała się być bardzo silna, a przebieg wykresu jest gładki i ma charakter spadkowy. Zmienna decyzyjna w takiej postaci oczywiście zostaje do późniejszych zastosowań dla modeli.

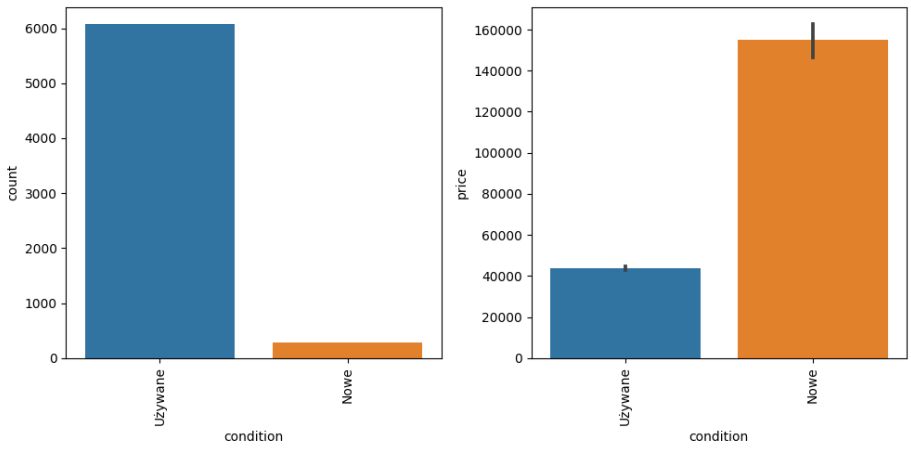
3.13 Kolor



Rysunek 18: Histogram koloru oraz wykres zależności ceny od koloru samochodu

Dla wielu kolorów samochodów, mamy bardzo mało danych. Tylko cztery spośród siedemnastu kolorów stanowią większość wśród danych. Pomimo więc wyraźnego zróżnicowania poziomu cen dla samochodów o różnych kolorach, zdecydowałem się odrzucić tę porcję danych z analizy.

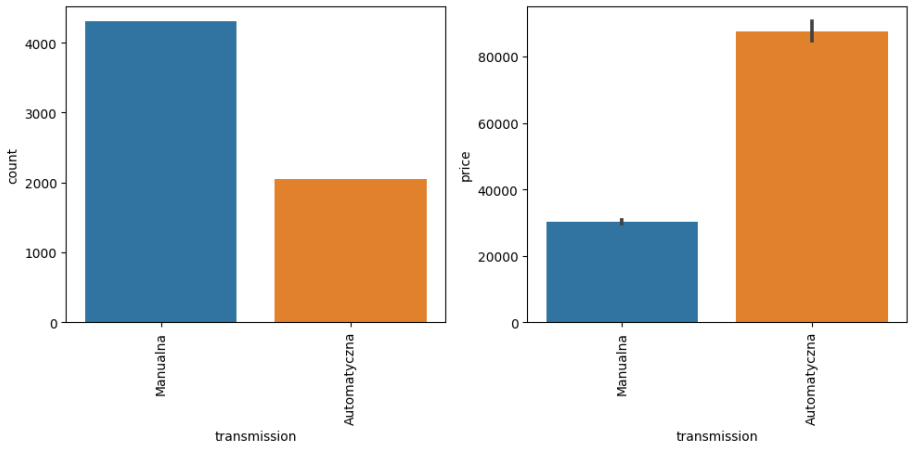
3.14 Stan



Rysunek 19: Histogram stanu oraz wykres zależności ceny od stanu samochodu

Mamy do czynienia z silną przewagą w ilości samochodów używanych nad nowymi. Nie możemy jednak z tego powodu po prostu usunąć informacji o stanie, gdyż z drugiej strony nowe samochody są zauważalnie dużo droższe od używanych. Ze względu na tę silną korelację - pozostawimy te dane.

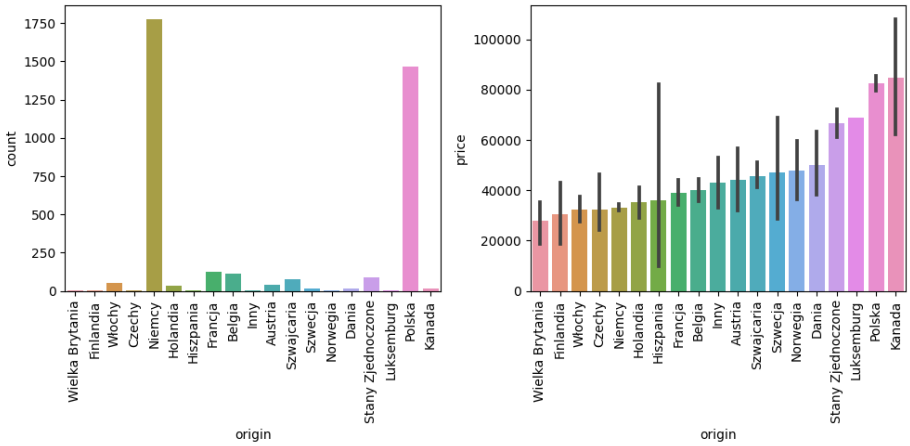
3.15 Skrzynia biegów



Rysunek 20: Histogram rodzajów skrzyni biegów oraz wykres zależności ceny od rodzaju skrzyni biegów

W tym przypadku mamy ponad dwukrotną przewagę w ilości samochodów z manualną skrzynią biegów w stosunku do samochodów z automatyczną skrzynią. Skrzynia automatyczna natomiast jest średnio trzykrotnie droższa od manualnej. Warto zatem zostawić te dane do nauki dla przyszłych modeli.

3.16 Pochodzenie



Rysunek 21: Histogram kraju pochodzenia oraz wykres zależności ceny od kraju pochodzenia samochodu

Prawie wszystkie samochody pochodzą z Polski lub Niemiec. Mamy niestety zbyt wiele różnych innych krajów z przypisaną do siebie małą ilością samochodów, przez co nie powinniśmy ufać zależnością przedstawionym na wykresie. Dodatkowo, tylko przy nieco ponad 4000 samochodów, mamy niepusty kraj pochodzenia, czyli w przypadku niecałych 40% przypadków kraj pochodzenia samochodu jest nieznany. Z obu tych względów rozsądnie będzie odrzucić tę porcję danych z dalszej analizy.

4 Końcowe przetwarzanie danych

4.1 Uzupełnienie wartości pustych

Do uzupełniania wartości pustych przyjęte zostały dwie strategie: po jednej dla danych ciągłych i kategorycznych.

- Dane ciągłe:

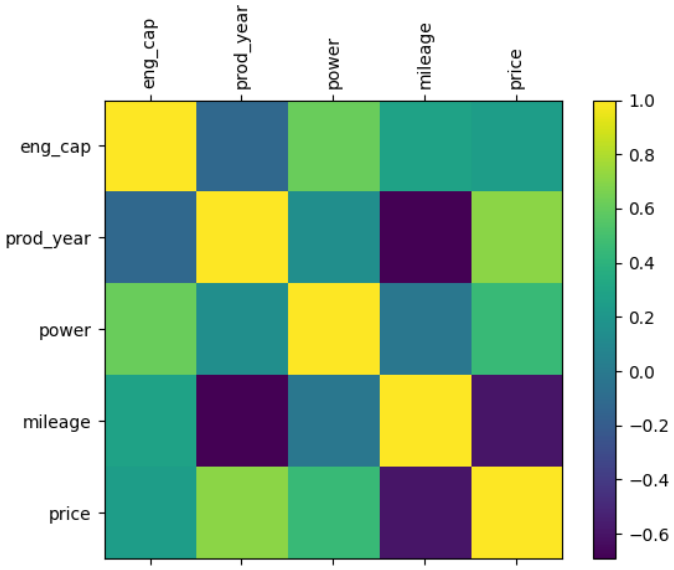
  tutaj strategia jest bardzo prosta - wszystkie dane są uzupełniane wartością średnią.

- Dane kategoryczne:

  strategia jest następująca - dla każdej kategorii liczone jest prawdopodobieństwo jej wystąpienia i dla każdych uzupełnianych danych losowana jest kategoria zgodnie z policzonym wcześniej prawdopodobieństwem.

Po uzupełnieniu danych, należy sprawdzić, czy nie ma żadnych duplikatów, które mogły powstać przy usuwaniu zmiennych decyzyjnych lub przy wypełnianiu pól pustych i ewentualnie takie powtórzenia usunąć.

Na tym etapie, po przeprowadzonej analizie, jako że dane są już prawie gotowe jako wejście dla modeli, przedstawię macierz korelacji dla przetworzonych danych. Celem łatwej interpretacji, na macierzy zostaną tylko wartości numeryczne (bez zmiennych kategorycznych):



Rysunek 22: Macierz korelacji zmiennych numerycznych

Jak widać na przedstawionej macierzy, pojemność silnika oraz moc są całkiem dobrze skorelowane z ceną, ale to z rokiem produkcji cena ma najwyższy współczynnik korelacji - co z resztą było widać na poprzednich wykresach - które były najmniej zaszumione i najgładsze właśnie dla roku produkcji. Pozostała reszta - czyli przebieg samochodu wykazał bardzo wysoki ujemny współczynnik korelacji - co również ma odzwierciedlenie w kształcie poprzednich wykresów, które dla przebiegu były również jednymi z najgładszych i najbardziej równomiernie rozłożonych - tyle że w przeciwieństwie do wcześniej wymieonionych - wykres przebiegu miał charakter spadkowy - stąd ujemna wartość współczynnika.

4.2 Kodowanie danych kategorycznych

Wszystkie dane kategoryczne muszą zostać odpowiednio zakodowane celem odpowiedniej ich interpretacji przez model. Na rzecz tego projektu, przetestowałem dwa sposoby kodowania: *one-hot* encoding oraz *label encoding* [6]. Pierwszy sposób dla jednej kolumny danych kategorycznych tworzy n kolumn, gdzie n jest liczbą wszystkich kategorii. Dla pojedynczej próbki tylko w jednej z tych kolumn może wystąpić wartość 1, w reszcie występuje 0 - stąd **one-hot**. Drugi sposób działa prościej - dla kolumny danych kategorycznych, przypisuje każdej kategorii liczbę (od 0 w górę) - zatem ilość kolumn się nie zmienia tylko zamiast kategorii jaka w tej kolumnie występowałą, zawarta jest od tej chwili odpowiadająca jej liczba.

Oba te sposoby kodowania dały bardzo podobne wyniki modeli, wybrałem więc label encoding - głównie dlatego, że niektórych kategorii było naprawdę dużo (np. generacji samochodu) - a mogłoby ich być jeszcze więcej dla nieco innych danych.

	generation	eng_cap	prod_year	power	mileage	condition	transmission	price
1221	17	1968.0	2022	150.0	32000	1	0	95000.0
1941	17	1968.0	2020	150.0	76020	1	0	86900.0
573	7	2996.0	2005	258.0	288858	1	1	32500.0
1511	14	1390.0	2007	122.0	180000	1	1	13400.0
1702	2	1229.0	2008	80.0	187000	1	1	14000.0

Rysunek 23: Reprezentacyjny wycinek danych po zastosowaniu label encodingu

4.3 Podział danych

Na tym etapie dane, przechowywane do tej pory w jednolitej formie, zostaną podzielone na dwa podzbiory: zbiór treningowy - dla nauki modelu, oraz zbiór testowy - do testowania modelu, w stosunku 3:1, gdyż po przeprowadzonych testach - taki właśnie stosunek dawał najlepsze rezultaty.

4.4 Normalizacja

Po zakodowaniu danych kategorycznych, należy znormalizować część zbioru danych, zawierającą zmienne decyzyjne - zarówno część testową jak i treningową. Polega to na przekształceniu każdej zmiennej losowej do zmiennej rozkładu normalnego o parametrach $\mu = 0$ oraz $\sigma = 1$. Wzorem, normalizacja przedstawia się następująco:

$$\overline{X} = \frac{X - \mu}{\sigma} \tag{1}$$

Operacja ta wykonywana jest po to, żeby wartości wszystkich zmiennych dezycyjnych zawierały się w podobnych przedziałach - aby żadna ze zmiennych nie wyhylała się wartościami ponad inne i żeby przez to model nie faworyzował jej wśród reszty [2].

Dlatego też normalizujemy tylko zmienne decyzyjne, a cena może pozostać nieznormalizowana, gdyż jest wyjściem modelu, a nie jego wejściem.

5 Wybór modeli

Proces obróbki danych został zakończony. Teraz nadszedł czas na wybór modeli, dla których dane te zostaną dopasowane. Na cel projektu rozważę następujące modele:

- Random Forest
- Decision Tree
- Gradient Boosting

5.1 Szukanie najlepszych parametrów modelu

Każdy model przyjmuje inne parametry, decydujące o poziomie jego dopasowania do danych.

Aby dobrać parametry, dające jak najlepsze wyniki, użyjemy tzw. Grid Search [5]. Polega to na tym, że dla zadanej siatki parametrów (różnych wartości podanych dla konkretnych parametrów modelu), sprawdzane są wszystkie możliwe kombinacje. Dodatkowo Grid Search przjmuje charakterystyczny parametr *cv* - cross-validation. Określa on na ile części zostaną podzielone podane dane, np. dla $cv = 5$, dane zostaną podzielone na 5 części, gdzie stosunek treningowych do testowych wynosi 4:1. Model zostanie przetestowany dla każdego możliwego podziału na dane testowe i treningowe - w tym wypadku będzie 5 możliwości takiego podziału.

5.2 Decision Tree

Drzewo decyzyjne [1] to jeden z podstawowych modeli uczenia maszynowego używanych do zadań klasyfikacji i regresji. Model ten działa poprzez rekurencyjne dzielenie danych na mniejsze podzbiory, tworząc strukturę drzewa, gdzie każdy węzeł odpowiada decyzji na podstawie wartości pewnej cechy.

Najważniejsze parametry drzewa decyzyjnego to:

- criterion: Funkcja oceny podziału, np. “squared_error”, “friedman_mse”, “absolute_error”, “poisson”
- splitter: Strategia użyta do wyboru podziału dla każdego węzła: “best” - najlepszy podział, “random” - najlepszy losowy podział

- max_depth: Maksymalna głębokość drzewa. Ograniczenie głębokości drzewa zapobiega jego nadmiernemu dopasowaniu (overfitting).
- max_features: Liczba cech rozważanych przy każdym podziale. Może być wartością absolutną, procentową lub "sqrt"(pierwiastek kwadratowy liczby cech) lub "log2".
- min_samples_split: Minimalna liczba próbek wymagana do podziału węzła. Większe wartości pomagają w uniknięciu nadmiernego dopasowania.
- min_samples_leaf: Minimalna liczba próbek, które muszą znajdować się w liściu. Wyższe wartości mogą poprawić ogólną stabilność modelu.

5.3 Random Forest

Las losowy [8] to złożony model uczenia maszynowego, który łączy wiele drzew decyzyjnych, aby poprawić dokładność i kontrolować nadmierne dopasowanie. Każde drzewo w lesie losowym jest trenowane na losowym podzbiore danych z użyciem losowych podzbiorów cech, co zwiększa różnorodność i ogólną wydajność modelu.

Najważniejsze parametry lasu losowego to:

- n_estimators: Liczba drzew w lesie. Większa liczba drzew zwykle poprawia dokładność modelu, ale zwiększa również czas obliczeń.
- max_depth: Maksymalna głębokość drzew. Może być używana do kontrolowania złożoności każdego pojedynczego drzewa.
- max_features: Podobnie jak w przypadku Decision Tree.
- min_samples_split i min_samples_leaf: Podobnie jak w drzewach decyzyjnych, parametry te pomagają kontrolować złożoność modeli i przeciwdziałać nadmiernemu dopasowaniu.

5.4 Gradient Boosting

Gradient Boosting [4] to zaawansowana technika uczenia maszynowego, która iteracyjnie łączy słabe modele, takie jak małe drzewa decyzyjne, aby tworzyć silny model. Każde kolejne drzewo koryguje błędy popełnione przez poprzednie drzewa, co prowadzi do coraz lepszej wydajności modelu.

Najważniejsze parametry gradientowego boostingu to:

- loss: Optymalizowana funkcja straty - model wykorzystuje tę funkcję, celem weryfikacji, czy w kolejnych iteracjach algorytmu, przewidywania modelu uległy poprawie.
- n_estimators: Liczba drzew w sekwencji. Większa liczba estymatorów może poprawić dokładność, ale zwiększa również ryzyko nadmiernego dopasowania.

- learning_rate: Szybkość uczenia, która skaluje wkład każdego drzewa. Mniejsza wartość learning_rate wymaga większej liczby estymatorów, aby osiągnąć ten sam poziom wydajności.
- max_depth: Maksymalna głębokość każdego drzewa. Kontroluje złożoność każdego modelu bazowego.
- subsample: Proporcja próbek używana do trenowania każdego drzewa. Wartości poniżej 1.0 mogą pomóc w redukcji wariancji i nadmiernego dopasowania.
- min_samples_split i min_samples_leaf: Parametry te, podobnie jak w drzewach decyzyjnych i lasach losowych, kontrolują minimalną liczbę próbek wymaganą do podziału węzła i minimalną liczbę próbek w liściu.

6 Dopasowywanie modelu

Używając wybranych modeli wraz z siatkami parametrów, używamy wprowadzoną wcześniej metodę Grid Search w celu jak najlepszego dopasowania modeli do danych.

Dla poszczególnych modeli, parametry prezentują się następująco:

- Decision Tree
  - splitter: ['best', 'random'],
  - max_depth: [None, 10, 20, 30],
  - min_samples_split: [2, 5, 10],
  - min_samples_leaf: [1, 2, 4],
  - max_features: ['sqrt', 'log2']
- Random Forest
  - n_estimators: [100, 200, 300],
  - max_depth: [None, 5, 10],
  - min_samples_split: [2, 5, 10]
- Gradient Boosting
  - loss: ['squared_error', 'huber', 'quantile'],
  - max_depth: [3, 5],
  - min_samples_split: [2, 5],
  - min_samples_leaf: [1, 4]

Dodatkowo każdy z modeli posiada parametr SEED - określający ziarno losowości - celem reprodukcji eksperymentów.

6.1 Metryki

Teraz czas przejść do metryk użytych aby opisać wyniki uzyskane przez modele.

6.1.1 Mean absolut error

Wzór:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

Jest to najbardziej podstawowa metryka, użyta do ocenienia modeli. Określa ona jak duży błąd jest popełniany przez model w predykowaniu ceny dla samochodu [7].

Model	Wynik
Decision Tree	9878.346356
Random Forest	7603.196992
Gradient Boosting	7681.683557

Tabela 2: MAE różnych modeli

Można więc, dla przykładu, stwierdzić, że cena samochodu predykowana przez Random Forest, wynosi: $cena\_auta \pm 7603.20 PLN$

6.1.2 Root mean squared error

Wzór:

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

Będzie to najważniejsza metryka naszych modeli - jej również używa Grid Search, więc to pod nią głównie model będzie dopasowany. Różni się ona od MAE m.in. tym, że jest wrażliwsza na wartości odstające - i jej wartość jest zawsze wyższa lub równa wartości MAE dla jednych danych. Z tego względu, aby oceniać nasz model też pod względem 'konsekwentnego' działania (czy nie robi pojedynczych, ale dużych błędów), użyjemy tej metryki [9].

Model	Wynik
Decision Tree	16654.488647
Random Forest	12982.622761
Gradient Boosting	13359.364812

Tabela 3: RMSE różnych modeli

6.2 Metryka niestandardowa

Ta metryka to właściwie procentowa wersja metryki RMSE, powstała po to, by klarownie przedstawić porównanie użytych modeli, za pomocą błędu względnego. Metryka ta, w przeciwieństwie do pozostałych, zwiększa swoją wartość (maksymalnie 1) wraz z polepszaniem się modelu. Jest ona liczona następująco:

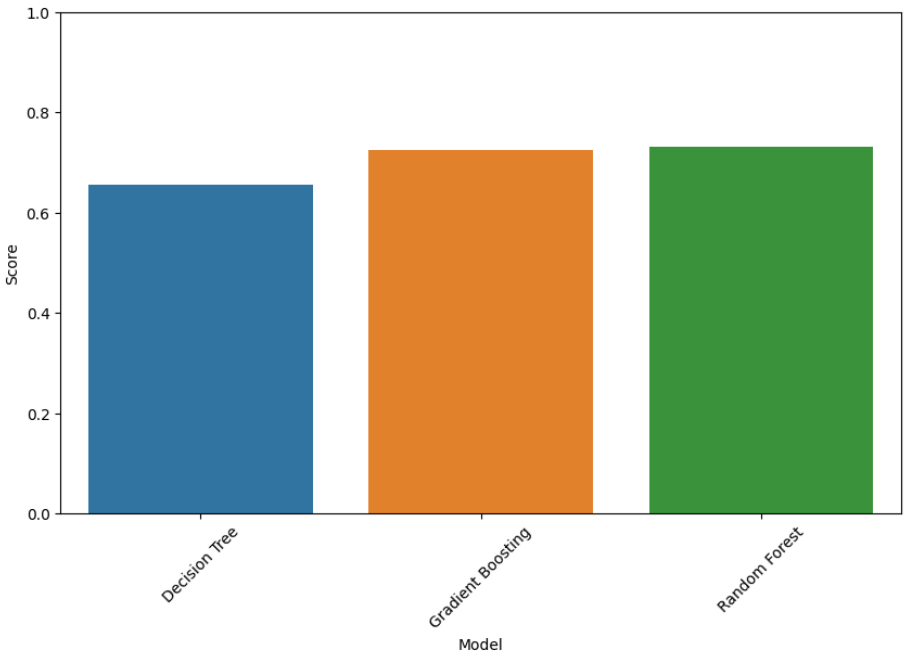
$$\text{METRYKA} = 1 - \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}}{\frac{1}{n}\sum_{i=1}^{n} y_i} \tag{2}$$

7 Wyniki

Używając wprowadzonej wyżej funkcji, wyniki uzyskane przez modele prezentują się następująco:

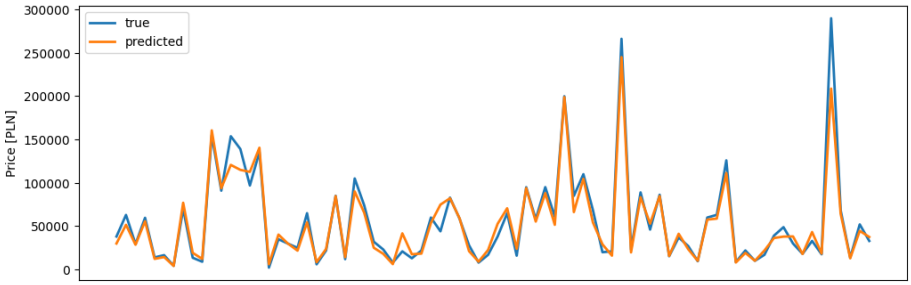
Model	Wynik
Decision Tree	0.656413
Gradient Boosting	0.724392
Random Forest	0.732164

Tabela 4: Końcowe metryki modeli



Rysunek 24: Diagram końcowych metryk modeli

Dodatkowo, celem demonstracji jak najlepszy z modeli, tj. Random Forest radzi sobie na danych testowych, poniżej widnieje wykres przedstawiający wycinek danych testowych, przedstawionych jako połączone punkty, reprezentujące cenę (true) oraz drugi wykres, który przedstawia jak model te dane predykuje (predicted).



Rysunek 25: Wycinek danych testowych i ich predykcja przez model

8 Wnioski

Model **Random Forest** uzyskał najlepszy wynik, osiągając około 73,22%. **Gradient Boosting** był niecały procent gorszy, natomiast **Decision Tree** aż 7,6% gorszy. Wynika to z faktu, że zarówno Random Forest, jak i Gradient Boosting są metodami opartymi na drzewach decyzyjnych. Random Forest tworzy wiele drzew decyzyjnych i łączy ich wyniki, co zwiększa stabilność i dokładność modelu. Gradient Boosting natomiast tworzy sekwencję drzew, z których każde kolejne drzewo koryguje błędy poprzednich, co także poprawia wydajność. Z tego powodu oba te modele zazwyczaj przewyższają pojedyncze drzewo decyzyjne. Pomimo tych różnic, ich wspólna podstawa w drzewach decyzyjnych sprawia, że wyniki są do siebie stosunkowo zbliżone.

Modele całkiem dobrze poradziły sobie predykowaniem użytych danych, jednak ze względu na wielowymiarowość i nieliniowe zależności danych - metryka modeli nie była na bardzo wysokim poziomie. Uważam jednak, że jak na stopień skomplikowania analizowanych danych - rezultaty są akceptowalne.

Dodatkowo, wyniki projektu ukazują, że to przebieg oraz rok produkcji samochodu mają największe znaczenie w kształtowaniu się jego ceny i właśnie na te aspekty należy w dużym stopniu zwracać uwagę przy wyborze auta.

Literatura

[1] Decision tree. https://scikit-learn.org/stable/modules/tree.html. Accessed: 2024-05-18.

[2] Features normalization. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Accessed: 2024-05-18.

[3] Forex-python. https://forex-python.readthedocs.io/en/latest/usage.html. Accessed: 2024-05-18.

[4] Gradient boosting. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html. Accessed: 2024-05-18.

[5] Grid search cv. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Accessed: 2024-05-18.

[6] Label encoding and one-hot encoding. https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/. Accessed: 2024-05-18.

[7] Mean absolute error. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html. Accessed: 2024-05-18.

[8] Random forest. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. Accessed: 2024-05-18.

[9] Root mean squared error. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.root_mean_squared_error.html. Accessed: 2024-05-18.

[10] Three sigma rule. https://andymath.com/normal-distribution-empirical-rule/. Accessed: 2024-05-18.