

Analiza i prognozowanie cen samochodów używanych na aukcjach z wykorzystaniem regresji

Zuzanna Jużyniec

zuzanna.juzyniec@student.pk.edu.pl

Rafał Matusiak

rafal.matusiak@student.pk.edu.pl

December 22, 2024

Contents

1 Abstract	2
2 Wprowadzenie	2
3 Zbiór danych	3
3.1 Wykres procentowej ilości brakujących wartości	4
3.2 Histogramy	4
3.3 Boxploty	6
3.4 Pairplot	9
3.5 Macierz korelacji	10
3.6 Wykres sprzedaży samochodów - Cena, Przebieg i Stan pojazdu dla 1000 losowych próbek	11
3.7 Wykres sprzedaży samochodów - Cena, Przebieg i Rok produkcji pojazdu dla 1000 losowych próbek	11
3.8 Liczba sprzedanych pojazdów w poszczególnych stanach USA	12
3.9 Top 10 najczęściej sprzedawanych marek samochodów	12
3.10 Liczba sprzedanych pojazdów według roku produkcji	13
3.11 Top 10 najlepiej sprzedających się typów nadwozia	13
3.12 Top 10 kolorów z największą liczbą sprzedanych pojazdów	14
4 Przegląd literatury	14
5 Motywacja	14
6 Ewaluacja	15
7 Zasoby	15
8 Zastosowane metody	15
9 Eksperyment	16
9.1 Preprocessing	16
9.2 Trening modelu	17
9.2.1 Podział danych na zbiory treningowe i testowe	17
9.2.2 Proces treningu modeli	18
9.2.3 Modele wykorzystane do analizy	19
9.2.4 Wyniki eksperymentów	19
9.2.5 Porównanie wyników na wykresach	21
9.2.6 Walidacja krzyżowa	23
9.2.7 Wnioski	24
9.2.8 Wnioski z analizy	25
9.3 Optymalizacja	26
9.3.1 Optymalizacja cech	26
9.3.2 Porównanie metod optymalizacji	28

9.3.3	Wyniki optymalizacji cech	28
9.3.4	Optymalizacja hiperparametrów	28
10	Podsumowanie	30

1 Abstract

W niniejszym projekcie zaprezentowano podejście do przewidywania cen samochodów używanych na podstawie historycznych danych aukcyjnych. Celem projektu jest zbudowanie modelu predykcyjnego, który na podstawie cech samochodów (takich jak marka, model, rok produkcji, stan techniczny, przebieg, itd.) prognozuje cenę sprzedaży pojazdu. Zastosowano kilka popularnych algorytmów uczenia maszynowego, takich jak regresja liniowa, lasy losowe, XGBoost oraz CatBoost. Dodatkowo, przeprowadzono optymalizację cech (RFE, PSO, Algorytmy Genetyczne) i hiperparametrów (GridSearchCV, RandomizedSearchCV, Optuna, Algorytmy Genetyczne). Projekt ma na celu rozwijanie umiejętności analizy danych oraz budowy modeli predykcyjnych.

2 Wprowadzenie

Przewidywanie cen samochodów na rynku wtórnym może przynieść korzyści zarówno kupującym, jak i sprzedającym, zapewniając obie strony bardziej przejrzystymi oraz sprawiedliwymi transakcjami. Dodatkowo, modele predykcyjne mogą wspomagać dealerów samochodowych, wypożyczalnie oraz platformy aukcyjne w dokładniejszej wycenie pojazdów, co pozwala na optymalizację procesów sprzedawczych. W ramach tego projektu, celem jest zbudowanie modelu przewidującego cenę samochodów na aukcjach na podstawie dostępnych danych.

3 Zbiór danych

Zbiór danych został pobrany ze strony <https://www.kaggle.com/datasets/tunguz/used-car-auction-prices?resource=download> i zawiera historyczne ceny sprzedaży samochodów na aukcjach, zebrane z zewnętrznych źródeł internetowych. Został zebrany w 2015 roku i nie był już aktualizowany.

Zbiór danych zawiera następujące kolumny:

- year - rok produkcji pojazdu;
- make - marka pojazdu (np. Toyota, Ford);
- model - model pojazdu (np. Corolla, Mustang);
- trim - wersja wyposażenia pojazdu (np. SE, Base);
- body - rodzaj nadwozia pojazdu (np. sedan, SUV);
- transmission - rodzaj skrzyni biegów (np. automatyczna, manualna);
- vin - numer identyfikacyjny pojazdu (VIN);
- state - stan (lokalizacja geograficzna) w USA, gdzie odbyła się sprzedaż;
- condition - stan pojazdu oceniany w skali od 1 do 5, gdzie wyższe wartości oznaczają lepszy stan techniczny (zmienna ciągła);
- odometer - przebieg pojazdu w milach;
- color - kolor karoserii pojazdu;
- interior - kolor wnętrza pojazdu;
- seller - nazwa lub identyfikator sprzedającego pojazd;
- mmr - wartość pojazdu według Manheim Market Report (MMR), używana jako wskaźnik wartości rynkowej;
- sellingprice - rzeczywista cena sprzedaży pojazdu na aukcji;
- saledate - data sprzedaży pojazdu.

W celu poznania danych zostało wykonane wiele wykresów.

3.1 Wykres procentowej ilości brakujących wartości

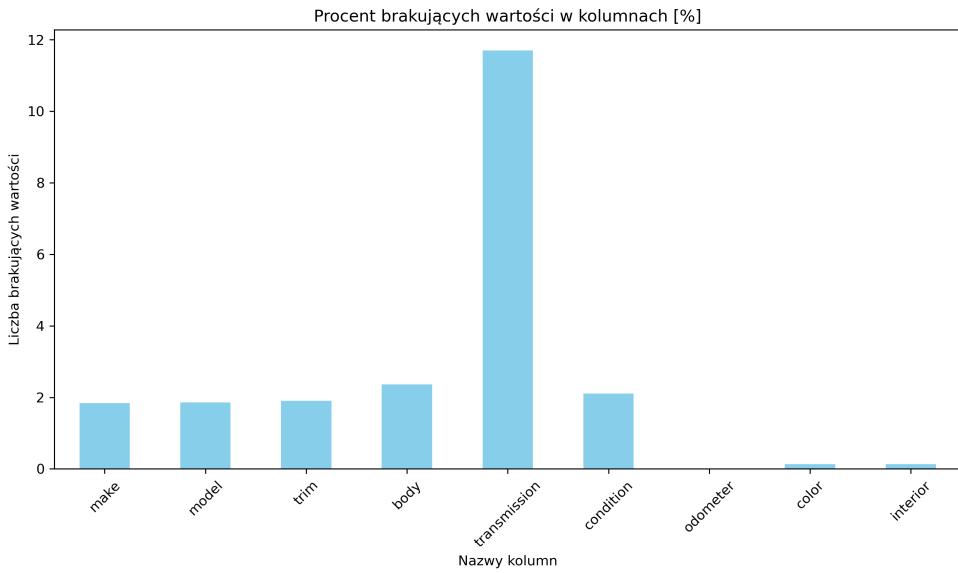


Figure 1: Histogramy zmiennych numerycznych.

Jak widać wartości NaN występują w 9 kolumnach, najczęściej w zmiennej 'transmission'.

3.2 Histogramy

Jak widać na Rys. 1, niektóre zmienne, takie jak sellingprice, mmr oraz odometer, charakteryzują się rozkładem długogonowym, co oznacza, że większość wartości koncentruje się w niższych przedziałach, jednak sporadycznie występują bardzo wysokie wartości odstające. Taki rozkład sugeruje, że typowe ceny sprzedaży, wartości rynkowe oraz przebiegi są zazwyczaj niskie, choć zdarzają się pojazdy o wyjątkowo wysokich cenach i dużym przebiegu.

Aby lepiej zobrazować te rozkłady, zastosowano przekształcenie logarytmiczne, co przedstawiono na Rys. 2 — po tej transformacji rozkłady stały się bardziej symetryczne i zbliżone do normalnego, co ułatwiło ich interpretację. Zmienna condition ma bardziej równomierny rozkład, wskazując na stabilny stan techniczny większości pojazdów, natomiast year wykazuje tendencję przesunięcia ku nowszym rocznikom, co sugeruje, że na aukcjach dominują młodsze samochody. Podczas dalszej analizy wyniknie, czy takie przekształcenie będzie konieczne w procesie modelowania, aby poprawić wyniki prognozowania, jednak na tym etapie pozostajemy przy badaniu oryginalnych danych.

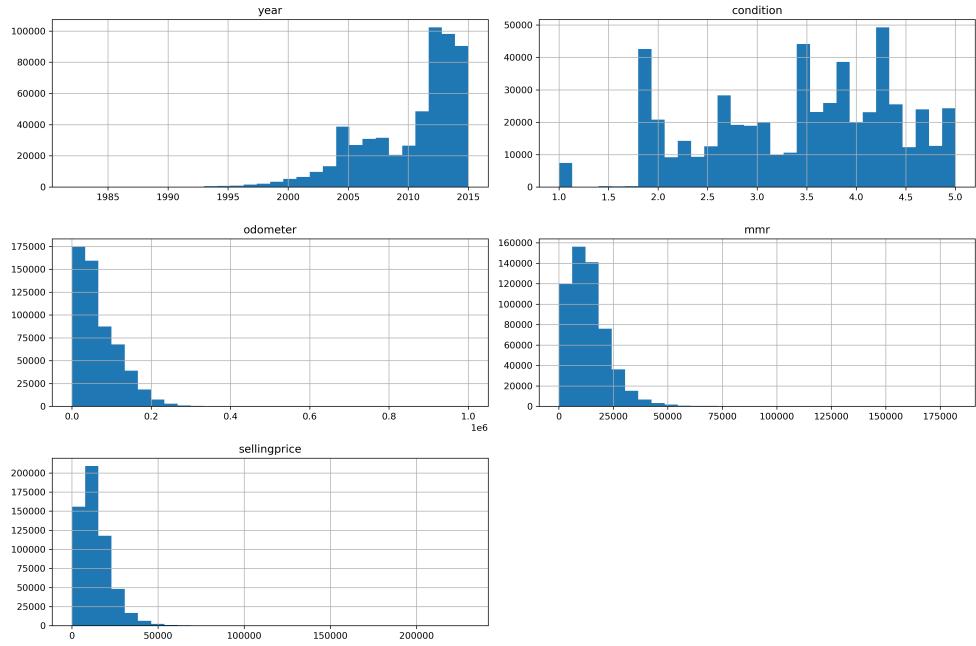


Figure 2: Histogramy zmiennych numerycznych.

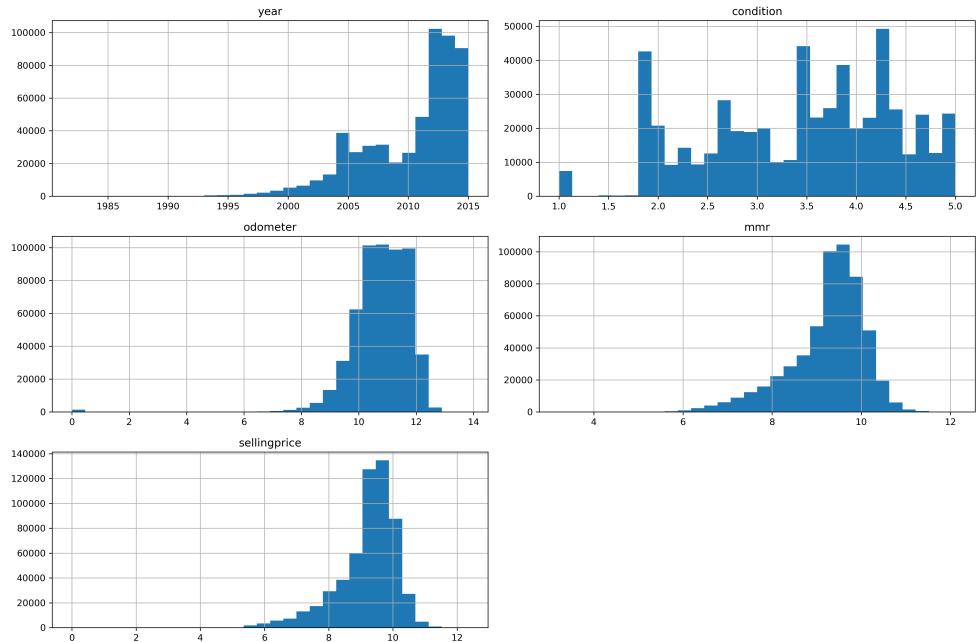


Figure 3: Histogramy zmiennych numerycznych po przekształceniu logarytmicznym.

3.3 Boxploty

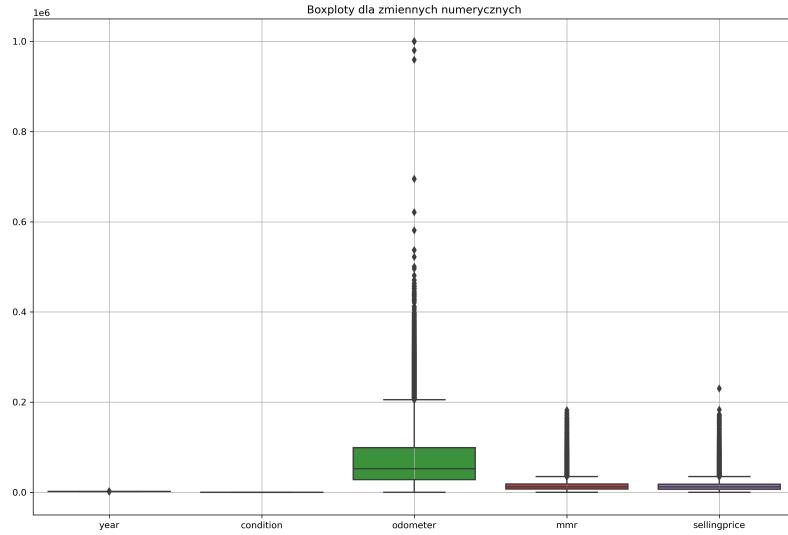


Figure 4: Boxploty dla wszystkich zmiennych numerycznych.

Na powyższym wykresie (Rys. 4) przedstawiono boxploty dla zmiennych numerycznych, jednak ze względu na dużą różnicę w wartościach pomiędzy zmiennymi, szczegóły niektórych cech są trudne do dostrzeżenia. Aby lepiej zobrazować rozkład wartości, w tym wartości odstające, zostaną przedstawione oddzielne boxploty dla każdej z cech numerycznych. Dodatkowo, wykresy te zostaną wygenerowane zarówno z uwzględnieniem danych odstających, jak i bez nich, aby zapewnić pełniejszą analizę danych.

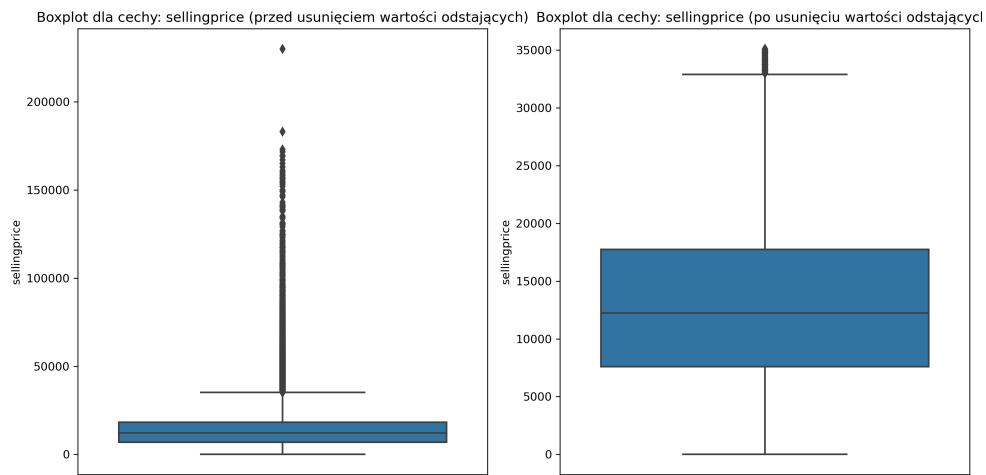


Figure 5: Boxploty dla zmiennej 'sellingprice'.

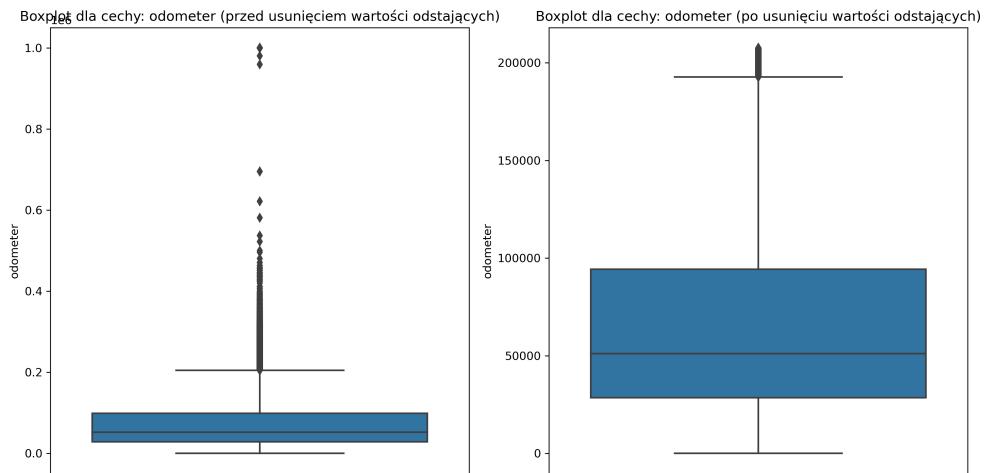


Figure 6: Boxploty dla zmiennej 'odometer'.

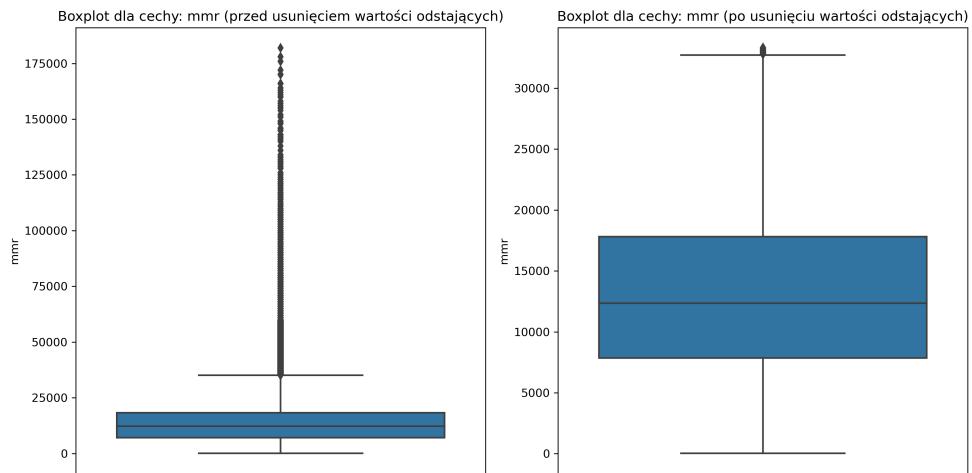


Figure 7: Boxploty dla zmiennej 'mmr'.

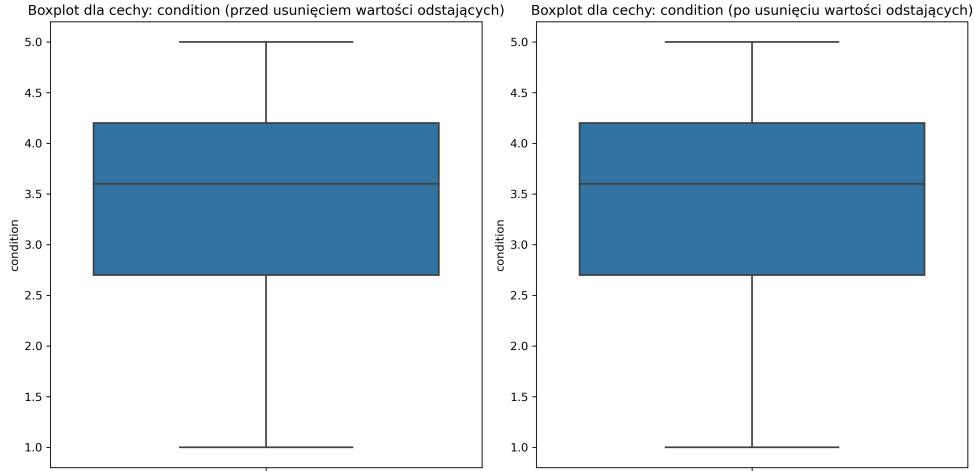


Figure 8: Boxploty dla zmiennej 'condition'.

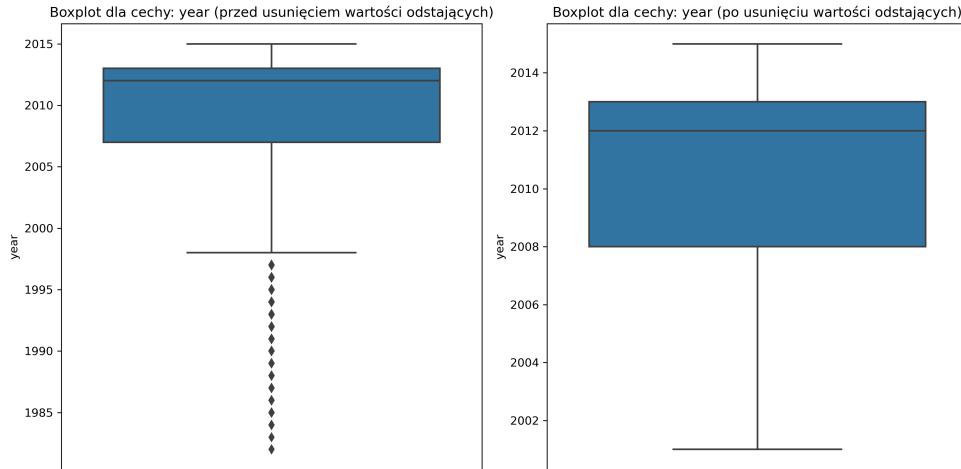


Figure 9: Boxploty dla zmiennej 'year'.

Wartości odstające mają największy wpływ na zmienne takie jak cena sprzedaży (sellingprice), przebieg (odometer) i wartość rynkowa (mmr), które charakteryzują się nielicznymi, ale bardzo skrajnymi wartościami odstającymi. Po ich usunięciu uzyskujemy bardziej klarowny obraz typowego rozkładu danych, co pozwala lepiej zrozumieć, jakie są rzeczywiste wartości dominujące w zbiorze danych.

W przypadku zmiennych rok produkcji (year) i stan techniczny (condition), wartości odstające mają mniejszy wpływ. Boxplot dla stanu technicznego jest bardzo jednorodny, co wskazuje, że ta zmienna nie zawiera wielu ekstremalnych wartości, a typowy stan techniczny pojazdów waha się od 2.5 do 4.5.

3.4 Pairplot

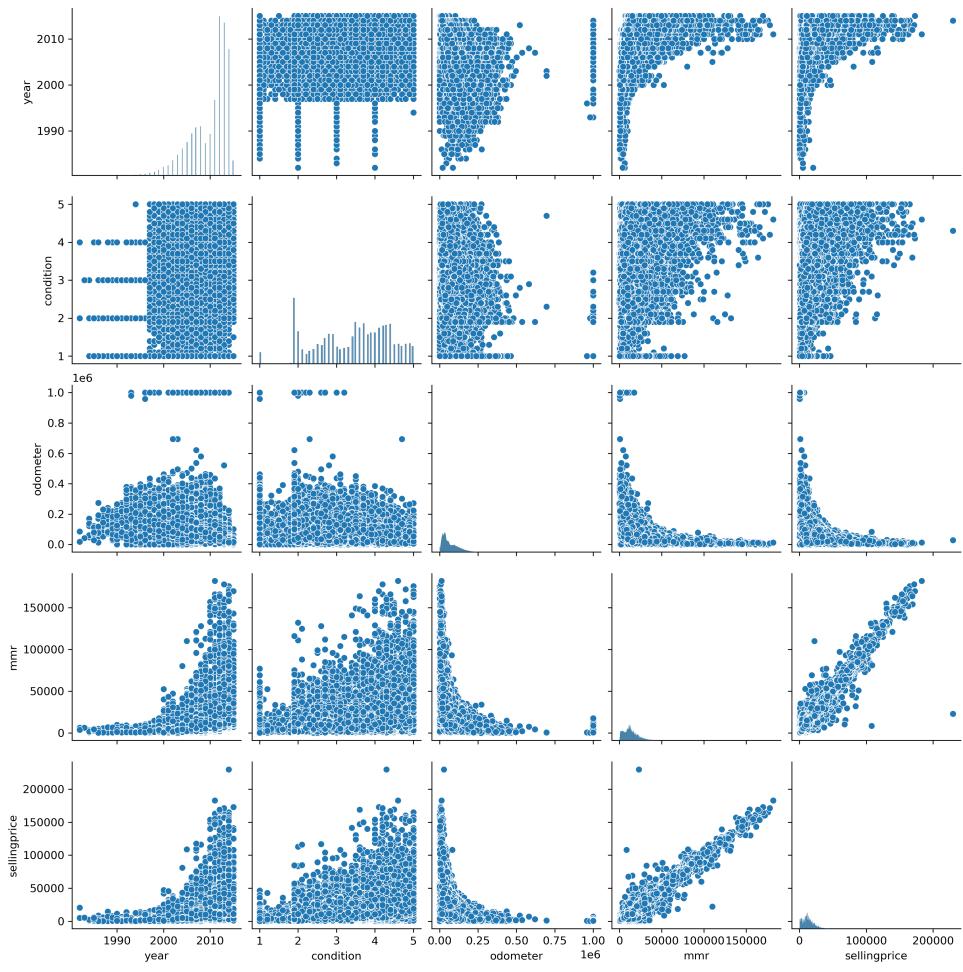


Figure 10: Pairplot.

Patrząc na pairplot, widzimy wyraźne zależności między niektórymi zmiennymi (np. cena sprzedaży a rok produkcji, przebieg czy wartość rynkowa). Aby lepiej zrozumieć, jak silne są te zależności, warto obliczyć współczynniki korelacji dla tych zmiennych. Pozwoli to na dokładniejszą ocenę siły tych relacji.

3.5 Macierz korelacji

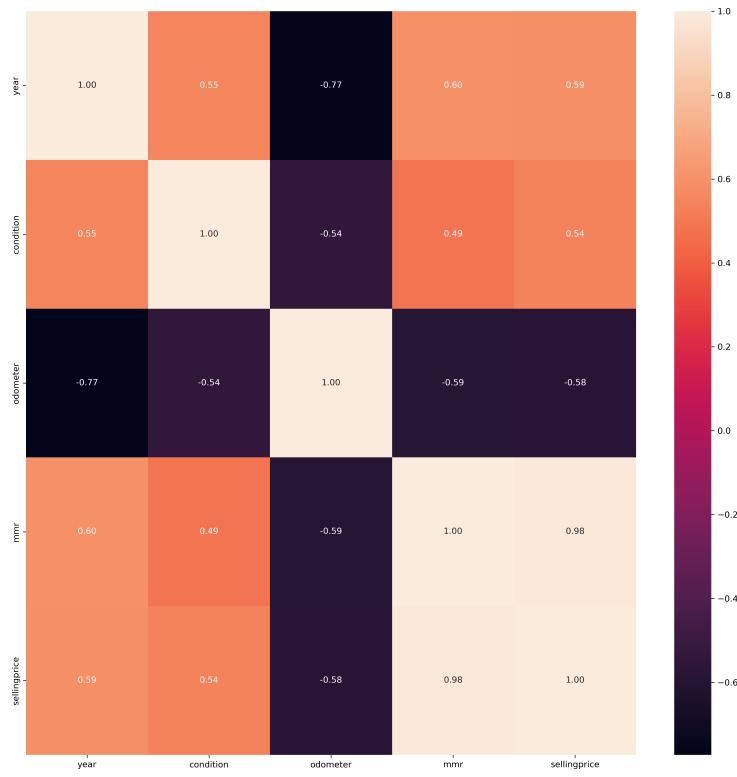


Figure 11: Macierz korelacji.

Rok produkcji (year) ma silną dodatnią korelację z ceną sprzedaży (0.59), co oznacza, że nowsze samochody zazwyczaj sprzedają się za wyższą cenę. Stan techniczny (condition) ma dodatnią korelację z ceną sprzedaży (0.54), co sugeruje, że samochody w lepszym stanie są droższe. Przebieg (odometer) wykazuje silną ujemną korelację z ceną sprzedaży (-0.58), co oznacza, że większy przebieg zmniejsza wartość samochodu. MMR (wartość rynkowa) ma bardzo silną dodatnią korelację z ceną sprzedaży (0.98), co wskazuje, że wartość rynkowa samochodu dobrze prognozuje jego cenę na aukcji.

3.6 Wykres sprzedaży samochodów - Cena, Przebieg i Stan pojazdu dla 1000 losowych próbek

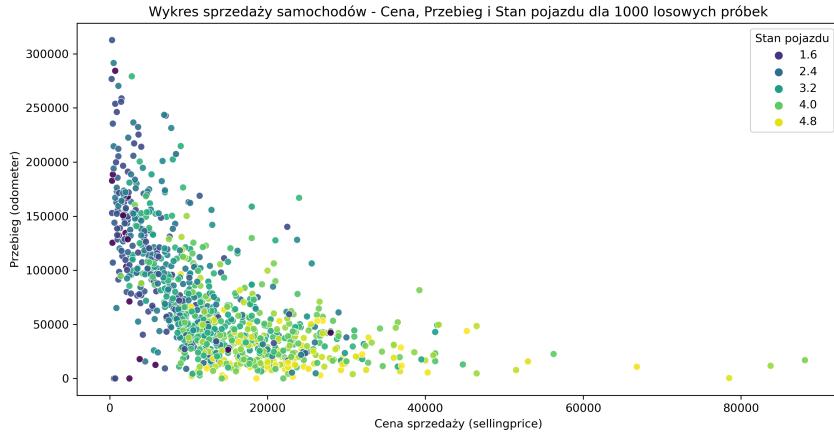


Figure 12: Wykres sprzedaży samochodów.

Na powyższym wykresie zbadano zależność między przebiegiem pojazdów a ceną sprzedaży, uwzględniając jednocześnie ich stan techniczny. Zauważalna jest silna ujemna zależność – samochody z większym przebiegiem są zazwyczaj tańsze. Lepszy stan techniczny pojazdu wpływa pozytywnie na cenę sprzedaży, co pozwala niektórym pojazdom osiągnąć wyższe ceny, nawet przy relatywnie dużym przebiegu. Najwyższe ceny uzyskują samochody w najlepszym stanie, podczas gdy pojazdy w najgorszym stanie sprzedają się za niższe kwoty, niezależnie od przebiegu.

3.7 Wykres sprzedaży samochodów - Cena, Przebieg i Rok produkcji pojazdu dla 1000 losowych próbek

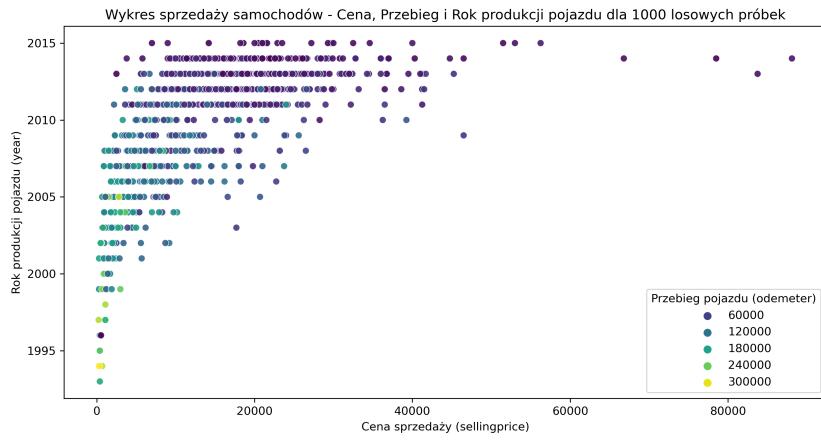


Figure 13: Wykres sprzedaży samochodów.

Wykres przedstawia zależność między rokiem produkcji a ceną sprzedaży samochodów, uwzględniając jednocześnie ich przebieg. Zauważalna jest wyraźna tendencja – nowsze samochody (z lat 2010–2015) osiągają wyższe ceny, podczas gdy starsze pojazdy (sprzed 2005 roku) sprzedają się za znacznie niższe kwoty. Przebieg również ma istotny wpływ – samochody

z mniejszym przebiegiem mają tendencję do wyższych cen, zwłaszcza w przypadku nowszych roczników. Samochody z większym przebiegiem, niezależnie od roku produkcji, zazwyczaj mają niższą wartość.

3.8 Liczba sprzedanych pojazdów w poszczególnych stanach USA



Figure 14: Liczba sprzedanych pojazdów w poszczególnych stanach USA.

Największą liczbę sprzedanych pojazdów odnotowano w stanach Floryda (FL) oraz Kalifornia (CA). Te dwa stany zdecydując wyróżniają się jako liderzy. W czołówce są także Teksas (TX), Pensylwania (PA) oraz Georgia (GA).

3.9 Top 10 najczęściej sprzedawanych marek samochodów

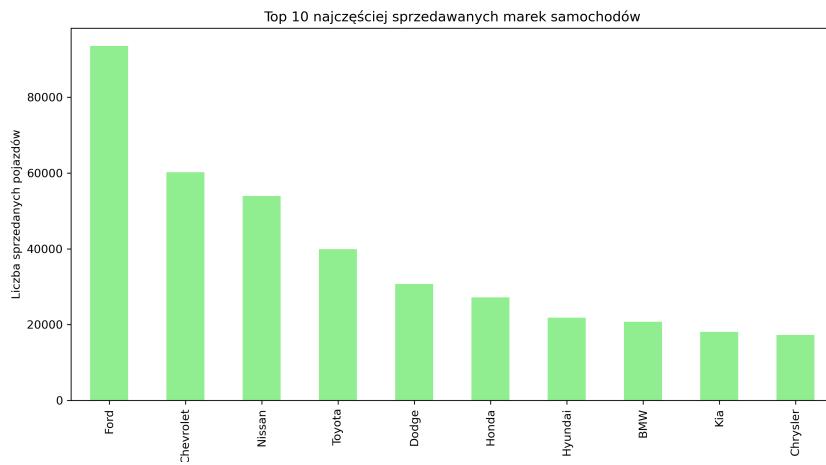


Figure 15: Top 10 najczęściej sprzedawanych marek samochodów

Ford zdecydując dominuje jako najczęściej sprzedawana marka, z wyraźną przewagą nad Chevroletem oraz Nissanem. Toyota oraz Dodge zamykają czołową piątkę. Wynik ten może odzwierciedlać popularność tych marek na rynku amerykańskim, gdzie dominują pojazdy amerykańskie i azjatyckie.

3.10 Liczba sprzedanych pojazdów według roku produkcji

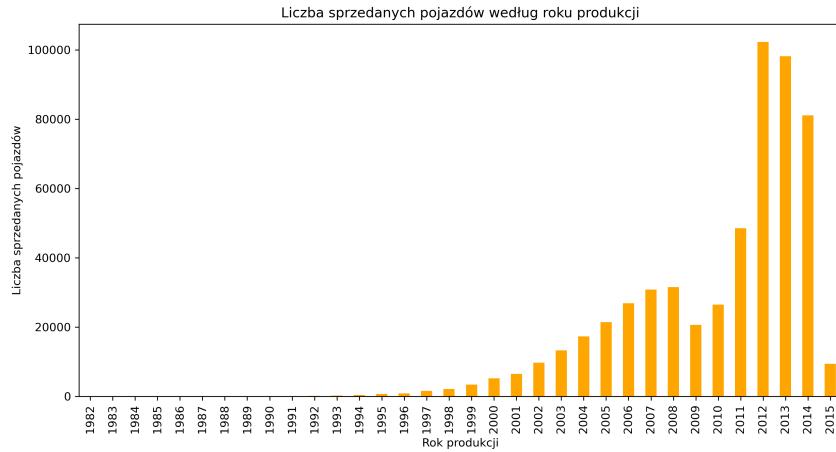


Figure 16: Liczba sprzedanych pojazdów według roku produkcji

Największa liczba sprzedanych pojazdów pochodzi z lat 2011-2014, co sugeruje, że rynek używanych samochodów obejmuje głównie nowsze modele, które mają jeszcze sporo lat eksploatacji przed sobą. Starsze roczniki, szczególnie te sprzed 2000 roku, są rzadziej spotykane, co jest zgodne z trendem zmniejszania się podaży starszych aut.

3.11 Top 10 najlepiej sprzedających się typów nadwozia

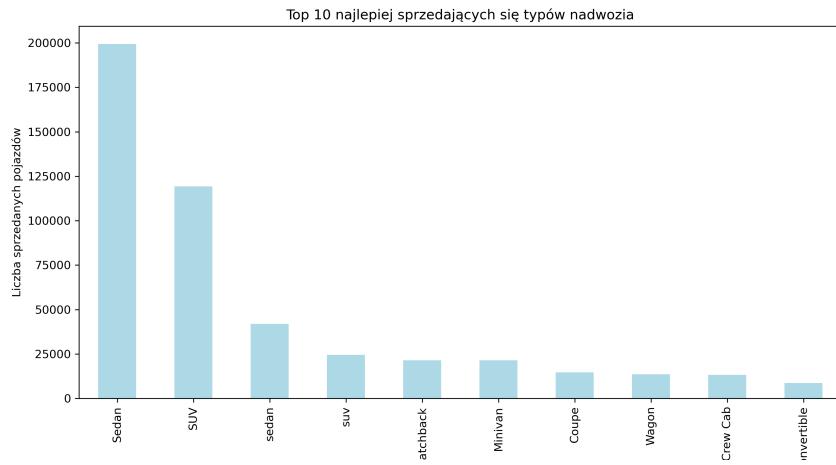


Figure 17: Top 10 najlepiej sprzedających się typów nadwozia

Sedan jest zdecydowanie najpopularniejszym typem nadwozia, z liczbą sprzedanych egzemplarzy wyraźnie przewyższającą inne typy. SUV zajmuje drugie miejsce, co świadczy o rosnącej popularności pojazdów tego segmentu. Inne typy nadwozia, takie jak hatchback czy minivan, również znajdują się w czołówce, ale z mniejszą liczbą sprzedanych egzemplarzy.

3.12 Top 10 kolorów z największą liczbą sprzedanych pojazdów

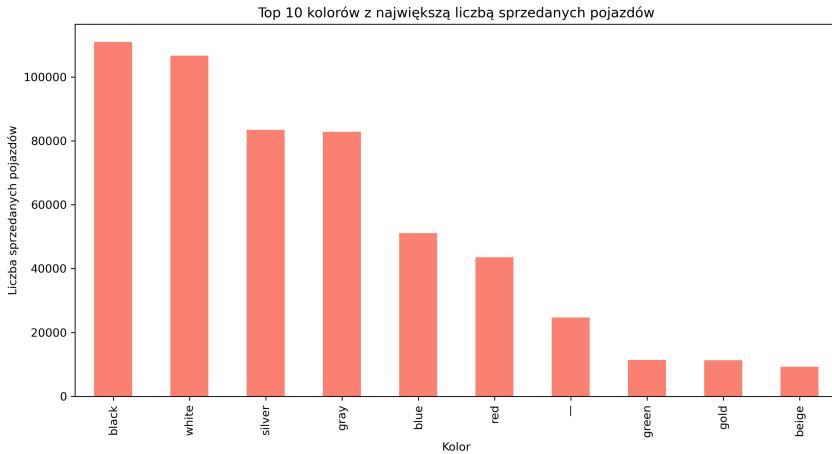


Figure 18: Top 10 kolorów z największą liczbą sprzedanych pojazdów

Kolory takie jak czarny, biały i srebrny są najczęściej wybierane przez nabywców, co nie jest zaskoczeniem, biorąc pod uwagę ich popularność na rynku. Są to kolory neutralne, które zazwyczaj lepiej utrzymują swoją wartość na rynku wtórnym. Inne kolory, takie jak niebieski i czerwony, również znajdują się w top 10, ale w mniejszych ilościach.

4 Przegląd literatury

Zbiór nie był wykorzystywany w publikacjach. Dostępne jest kilka notatników z przykładowymi podejściami do problemu.

- Notatnik Kaggle [2] autorstwa Zabihullah18 dotyczy predykcji cen samochodów za pomocą uczenia maszynowego. Zawiera kroki takie jak wstępne przetwarzanie danych, wybór cech i trenowanie modeli. Używa algorytmu regresji liniowej do przewidywania cen na podstawie danych o samochodach (np. marka, model, przebieg). Wyniki są oceniane przy użyciu wskaźników takich jak błąd średniokwadratowy, który pozwala mierzyć dokładność prognoz.
- Notatnik na Kaggle [1] dotyczący predykcji cen używanych samochodów autorstwa Midoel3ila wykorzystuje techniki uczenia maszynowego do przewidywania cen samochodów na podstawie ich cech. Proces obejmuje wstępne przetwarzanie danych, eksplorację cech takich jak marka, model, przebieg i wiek pojazdu, a następnie budowanie modeli predykcyjnych. Algorytmy takie jak regresja liniowa i lasy losowe są stosowane do trenowania modelu. Notatnik ocenia dokładność przewidywań za pomocą miar takich jak błąd średniokwadratowy.

5 Motywacja

Projekt dotyczący prognozowania cen samochodów używanych jest interesujący, ponieważ może przynieść korzyści zarówno kupującym, jak i sprzedającym, pomagając przewidzieć realistyczne ceny rynkowe. W społeczeństwie, gdzie kupno używanych pojazdów jest powszechnie, dokładne modele predykcyjne mogą wspierać transparentność i zaufanie w transakcjach. W przemyśle, takie prognozy mogą być użyteczne dla wypożyczalni samochodów, dealerów oraz platform aukcyjnych, pomagając w lepszym oszacowaniu wartości pojazdów i optymalizacji strategii sprzedaży.

6 Ewaluacja

Oczekiwany model będzie w stanie dokładnie przewidzieć ceny samochodów używanych na aukcjach na podstawie dostępnych danych. Satysfakcjonujący wynik to taki, w którym błąd predykcji (np. błąd średniokwadratowy) będzie niski, co oznacza, że przewidywane ceny są bliskie rzeczywistym cenom sprzedaży. Osiągniety wynik, powinien być wystarczająco precyzyjny, aby umożliwić zastosowanie modelu w rzeczywistych warunkach — np. na platformach aukcyjnych lub w firmach dealerskich — gdzie może pomóc w wycenie pojazdów. Oczywiście projekt ma charakter wyłącznie edukacyjny i nie jest przeznaczony do komercyjnego wykorzystania ani sprzedaży. Celem jest nauka i zrozumienie metod prognozowania cen samochodów używanych z wykorzystaniem modeli uczenia maszynowego. Chociaż dążymy do uzyskania dokładnych wyników, system ten nie będzie wdrażany w żadnym rzeczywistym środowisku komercyjnym. Skupiamy się na poszerzaniu wiedzy i umiejętności w zakresie analizy danych oraz budowania modeli predykcyjnych.

7 Zasoby

W projekcie zostanie wykorzystany język programowania Python oraz środowisko Jupyter Notebook do analizy danych i budowy modeli predykcyjnych. Python oferuje bogaty zestaw bibliotek do uczenia maszynowego, takich jak scikit-learn do budowy i oceny modeli, pandas do manipulacji danymi, oraz numpy do obliczeń numerycznych. Dodatkowo, zosatnie wykorzystana biblioteka matplotlib i seaborn do wizualizacji danych, co pozwoli na lepsze zrozumienie zależności i wzorców w zebranych informacjach.

8 Zastosowane metody

Do analizy i budowy modelu predykcyjnego zastosowano szereg popularnych metod uczenia maszynowego, w tym:

Regresja liniowa – metoda wykorzystywana do szacowania zależności między zmiennymi. Jest to podstawowy model regresyjny, który może służyć jako punkt odniesienia.

Lasy losowe – ensemble learning method, który wykorzystuje zestaw drzew decyzyjnych, gdzie każdy z nich uczy się na losowej próbce danych.

Histogram-based Gradient Boosting Regressor – algorytm oparty na metodzie gradient boosting, szczególnie przydatny przy pracy z dużymi zbiorami danych.

XGBoost – jeden z najpopularniejszych algorytmów w zakresie boostingu, znany ze swojej efektywności i szybkości działania, szczególnie w przypadku dużych zbiorów danych.

CatBoost – metoda gradient boosting opracowana z myślą o automatyzacji obsługi zmiennych kategorycznych, co czyni ją szczególnie skuteczną w tego rodzaju zadaniach.

Voting Regressor – metoda łączenia wyników z kilku różnych modeli, przy czym finalna prognoza jest średnią z prognoz wszystkich modeli.

Ponadto, przeprowadzono optymalizację hiperparametrów oraz cech w celu uzyskania jak najlepszych wyników z tych algorytmów.

9 Eksperyment

9.1 Preprocessing

W celu przygotowania danych do modelowania i analizy predykcyjnej, przeprowadzono zaawansowany preprocessing, który obejmował różne etapy mające na celu poprawienie jakości i spójności zbioru danych. Szczegółowe kroki opisano poniżej.

1. Obliczenie wieku samochodu (car_age) Wiek samochodu w momencie sprzedaży został obliczony na podstawie kolumny `saledate` poprzez odjęcie od roku sprzedaży (`sale_year`) roku produkcji pojazdu (`year`). Dodatkowo, z tej samej kolumny (`saledate`) wyodrębniono dodatkowe cechy, takie jak:

- **Miesiąc sprzedaży (sale_month),**
- **Rok sprzedaży (sale_year),**
- **Godzina sprzedaży (sale_hour),**
- **Minuta sprzedaży (sale_minute).**

Po tej transformacji przeanalizowano sprzedaż samochodów w różnych miesiącach i dniach tygodnia. Wyniki wskazują, że najczęściej sprzedaż miało miejsce w styczniu (140,815 samochodów) oraz lutym (163,054 samochody). Z kolei analiza sprzedaży według dnia tygodnia wykazała, że transakcje najczęściej odbywają się na początku tygodnia, zwłaszcza we wtorki (180,158 sprzedaży), natomiast najmniej w niedziele (11,868 sprzedaży).

Podczas analizy wieku pojazdów wykryto, że 201 rekordów miało ujemne wartości w kolumnie `car_age`. Można zatem założyć, że taka liczba samochodów została zakupiona w przedsprzedaży.

2. Ekstrakcja informacji z numeru VIN Z numeru VIN pojazdu przy użyciu biblioteki `vininfo` wyodrębniono informacje o kraju produkcji (`vin_country`). Na tym etapie zrezygnowano z pozyskiwania dodatkowych danych, takich jak typ pojazdu, ponieważ były one już zawarte w innych kolumnach (`model`, `body`). Po wyodrębnieniu kraju, kolumna `vin` została usunięta, gdyż nie była już potrzebna do dalszej analizy.

3. Usunięcie wartości odstających W celu usunięcia wartości odstających przeprowadzono analizę cech numerycznych (`year`, `condition`, `odometer`, `mmr`, `sellingprice`) za pomocą metody IQR (Interquartile Range). Wartości spoza zakresu $[Q1 - 1.5 * IQR, Q3 + 1.5 * IQR]$ zostały usunięte. Łącznie usunięto około 8.3% danych, co było akceptowalne, ponieważ nie przekraczało 10% zbioru.

4. Przekształcenie zmiennych kategorycznych na wartości liczbowe Wszystkie zmienne kategoryczne (`make`, `model`, `trim`, `body`, `transmission`, `state`, `color`, `interior`, `seller`, `vin_country`) zostały przekonwertowane na wartości liczbowe. Dla kolumny `transmission` zastosowano kodowanie binarne: 0 dla manualnej skrzyni biegów oraz 1 dla automatycznej. Dla pozostałych kolumn zastosowano "label encoding" co w kolejnych etapach może ograniczyć możliwość wykorzystania niektórych modeli. Przy tak dużych danych i wielu kategoriach niemożliwe było zastosowanie one-hot encodingu.

5. Sprawdzenie brakujących wartości Po zakończeniu wcześniejszych kroków przeanalizowano brakujące wartości w każdej z kolumn. Wyniki prezentowały się następująco:

year	0.000000%
make	1.843378%
model	1.860915%
trim	1.906011%
body	2.361263%
transmission	11.695010%
state	0.000000%
condition	2.110553%
odometer	0.016821%
color	0.134035%
interior	0.134035%
seller	0.000000%

```

mmr           0.000000%
sellingprice 0.000000%
car_age       0.000000%
sale_year     0.000000%
sale_month    0.000000%
sale_hour     0.000000%
sale_minute   0.000000%
sale_weekday  0.000000%
vin_country   0.000000%

```

6. Uzupełnianie brakujących wartości

Do imputacji brakujących danych zastosowano trzy metody:

- Średnia dla danych numerycznych oraz moda dla danych kategorycznych.
- Mediana dla danych numerycznych oraz moda dla danych kategorycznych.
- Imputacja metodą k-Nearest Neighbors (`KNNImputer`) z 3 sąsiadami.

7. Skalowanie i standaryzacja danych

W celu ujednolicenia wartości numerycznych zastosowano dwa podejścia:

- Skalowanie Min-Max w zakresie [0, 1]
- Standaryzacja do średniej 0 i odchylenia standardowego 1

Dla każdego z trzech zbiorów (imputacja średnią, medianą, KNN) zastosowano skalowanie i standaryzację, co doprowadziło do utworzenia sześciu różnych zestawów danych:

```

mean_min_max.csv
median_min_max.csv
knn_min_max.csv
mean_standard.csv
median_standard.csv
knn_standard.csv

```

8. Zapis przetworzonych danych

Wszystkie powyższe zbiory zostały zapisane jako pliki CSV. Zbiory te są gotowe do użycia w modelach predykcyjnych i zapewniają lepszą jakość danych, co przekłada się na większą dokładność modeli.

9.2 Trening modelu

9.2.1 Podział danych na zbiory treningowe i testowe

Dane podzielono na zbiory treningowe i testowe w proporcji **80:20**. Proces ten zrealizowano w sposób manualny, zgodnie z zaimplementowanym algorytmem, zamiast korzystania z gotowych funkcji. Podział przebiegał według następujących kroków:

1. Dane zostały wczytane i wymieszane przy użyciu funkcji `shuffle`, co umożliwiło losowe rozłożenie próbek, przy jednoczesnym zapewnieniu powtarzalności wyników dzięki zastosowaniu parametru `random_state=42`.
2. Na podstawie zdefiniowanego indeksu, wyznaczono dwie części zbioru:
 - Zbiór treningowy: zawierający pierwsze 80% obserwacji,
 - Zbiór testowy: obejmujący pozostałe 20%.

Podziału dokonano niezależnie dla każdego z sześciu przetworzonych zbiorów danych:

- `mean_min_max`,
- `median_min_max`,
- `knn_min_max`,
- `mean_standard`,

- median_standard,
- knn_standard.

Każdy zbiór danych zawierał **21** cech oraz około **409,778 obserwacji** w zbiorze treningowym i **102,445 obserwacji** w zbiorze testowym.

9.2.2 Proces treningu modeli

1. **Przygotowanie danych:** Zmienną objaśnianą była `sellingprice`, natomiast zmiennymi objaśniającymi pozostałe kolumny w zbiorze danych.
2. **Trenowanie modeli:** Modele trenowano za pomocą metody `fit()` na zbiorze treningowym, a predykcje generowano na zbiorze testowym za pomocą `predict()`.
3. **Ocena wyników:** Modele oceniano przy użyciu następujących metryk:
 - **Mean Absolute Error (MAE):** średni błąd bezwzględny,
 - **Root Mean Square Error (RMSE):** pierwiastek z błędu średniokwadratowego,
 - **R²:** współczynnik determinacji,
 - **Mean Absolute Percentage Error (MAPE):** średni błąd względny, wyrażony w procentach.

Dodatkowo zarejestrowano czas trenowania każdego modelu.

9.2.3 Modele wykorzystane do analizy

W eksperymencie wykorzystano następujące modele regresji:

- **Random Forest Regressor,**
- **Linear Regression,**
- **Histogram-based Gradient Boosting Regressor,**
- **XGBoost Regressor,**
- **CatBoost Regressor,**
- **Voting Regressor** składający się z **Linear Regression**, **Histogram-based Gradient Boosting Regressor**, **Elastic Net**,
- **Stacking Regressor** składający się z **Linear Regression**, **Histogram-based Gradient Boosting Regressor**, **Elastic Net**.

Model **Support Vector Regression (SVR)** został wykluczony ze względu na problemy z wydajnością przy dużych zbiorach danych oraz trudności w przetwarzaniu zmiennych kategorycznych (po zastosowaniu *label encoding*). W uczeniu zespołowym nie wykorzystano także algorytmu Random Forest ze względu na problemy z wydajnością.

9.2.4 Wyniki eksperymentów

Wyniki dla każdego modelu i zbioru danych zostały przedstawione w Tabeli 1.

Table 1: Wyniki dla każdego modelu i zbioru danych.

ID	Zbiór danych	Model	MAE	RMSE	R ²	MAPE	Czas (s)
0	mean_min_max	RandomForest	0.026047	0.039195	0.968885	12.421442	377.814284
1	mean_min_max	LinearRegression	0.028074	0.042012	0.964252	14.008762	0.405149
2	mean_min_max	HistGradientBoostingRegressor	0.025588	0.038294	0.970300	12.207046	11.084602
3	mean_min_max	XGBoost	0.024886	0.037412	0.971652	11.689896	3.698308
4	mean_min_max	CatBoost	0.024627	0.036911	0.972407	11.564949	70.450874
5	mean_min_max	Voting Regressor	0.065400	0.082988	0.860514	49.292124	8.702858
6	mean_min_max	Stacking Regressor	0.025575	0.038244	0.970378	12.154732	53.870504
7	median_min_max	RandomForest	0.026043	0.039188	0.968897	12.416976	257.189253
8	median_min_max	LinearRegression	0.028074	0.042012	0.964252	14.008762	0.354580
9	median_min_max	HistGradientBoostingRegressor	0.025555	0.038270	0.970337	12.196443	9.029640
10	median_min_max	XGBoost	0.024845	0.037338	0.971763	11.681776	3.241030
11	median_min_max	CatBoost	0.024592	0.036882	0.972450	11.548103	68.110814
12	median_min_max	Voting Regressor	0.065387	0.082980	0.860542	49.283282	8.520290
13	median_min_max	Stacking Regressor	0.025535	0.038212	0.970427	12.132452	50.229682
14	knn_min_max	RandomForest	0.026032	0.039191	0.968891	12.420795	255.567970
15	knn_min_max	LinearRegression	0.028098	0.042036	0.964211	14.004907	0.364529
16	knn_min_max	HistGradientBoostingRegressor	0.025589	0.038308	0.970279	12.201167	10.394278
17	knn_min_max	XGBoost	0.024855	0.037345	0.971753	11.641967	3.362012
18	knn_min_max	CatBoost	0.024622	0.036968	0.972321	11.566808	74.796223
19	knn_min_max	Voting Regressor	0.065349	0.082936	0.860688	49.191944	11.566812
20	knn_min_max	Stacking Regressor	0.025573	0.038258	0.970356	12.151640	65.034105
21	mean_standard	RandomForest	0.117083	0.176223	0.968908	78.383993	260.279401
22	mean_standard	LinearRegression	0.126269	0.188958	0.964252	85.846146	0.339964
23	mean_standard	HistGradientBoostingRegressor	0.115087	0.172232	0.970300	78.263048	10.777526
24	mean_standard	XGBoost	0.111790	0.167926	0.971767	75.951406	3.118486
25	mean_standard	CatBoost	0.110740	0.166021	0.972404	75.029054	73.516604
26	mean_standard	Voting Regressor	0.220150	0.286456	0.917844	74.406331	10.642051
27	mean_standard	Stacking Regressor	0.115055	0.172009	0.970377	78.510305	59.476473
28	median_standard	RandomForest	0.117069	0.176152	0.968933	78.477672	243.033475
29	median_standard	LinearRegression	0.126269	0.188958	0.964252	85.846146	0.304093
30	median_standard	HistGradientBoostingRegressor	0.114941	0.172125	0.970337	78.244558	9.704722
31	median_standard	XGBoost	0.111745	0.167936	0.971763	75.989966	3.438448
32	median_standard	CatBoost	0.110640	0.166014	0.972406	74.672273	68.956737
33	median_standard	Voting Regressor	0.220089	0.286425	0.917862	74.362599	8.220672
34	median_standard	Stacking Regressor	0.114877	0.171866	0.970427	78.481290	51.174468
35	knn_standard	RandomForest	0.117061	0.176270	0.968891	78.289942	247.432862
36	knn_standard	LinearRegression	0.126377	0.189066	0.964211	85.749496	1.059998
37	knn_standard	HistGradientBoostingRegressor	0.115077	0.172158	0.970326	78.214080	11.462982
38	knn_standard	XGBoost	0.111789	0.167967	0.971753	76.288017	2.724514
39	knn_standard	CatBoost	0.110701	0.166215	0.972339	74.427258	49.833858
40	knn_standard	Voting Regressor	0.220160	0.286490	0.917813	74.165903	8.574647
41	knn_standard	Stacking Regressor	0.115046	0.172042	0.970355	78.510303	66.176187

Kluczowe spostrzeżenia:

- **CatBoost** osiągnął najlepsze wyniki na większości zbiorów danych, minimalizując wartości MAE i RMSE, a także osiągając najwyższy R^2 .
- **XGBoost** uzyskał porównywalne wyniki, ale z nieco krótszym czasem trenowania.
- **Random Forest** wymagał znacznie więcej czasu obliczeniowego, co czyniło go mniej wydajnym w porównaniu do gradientowego boostingu.
- **Linear Regression** była najszybsza, ale jej dokładność pozostawiała wiele do życzenia w porównaniu do bardziej zaawansowanych modeli.

Table 2: Wyniki dla najlepszych modeli na różnych zbiorach danych.

Zbior danych	Model	MAE	RMSE	R^2	MAPE	Czas (s)
median_min_max	CatBoost	0.024592	0.036882	0.97245	11.548	68.110814

9.2.5 Porównanie wyników na wykresach

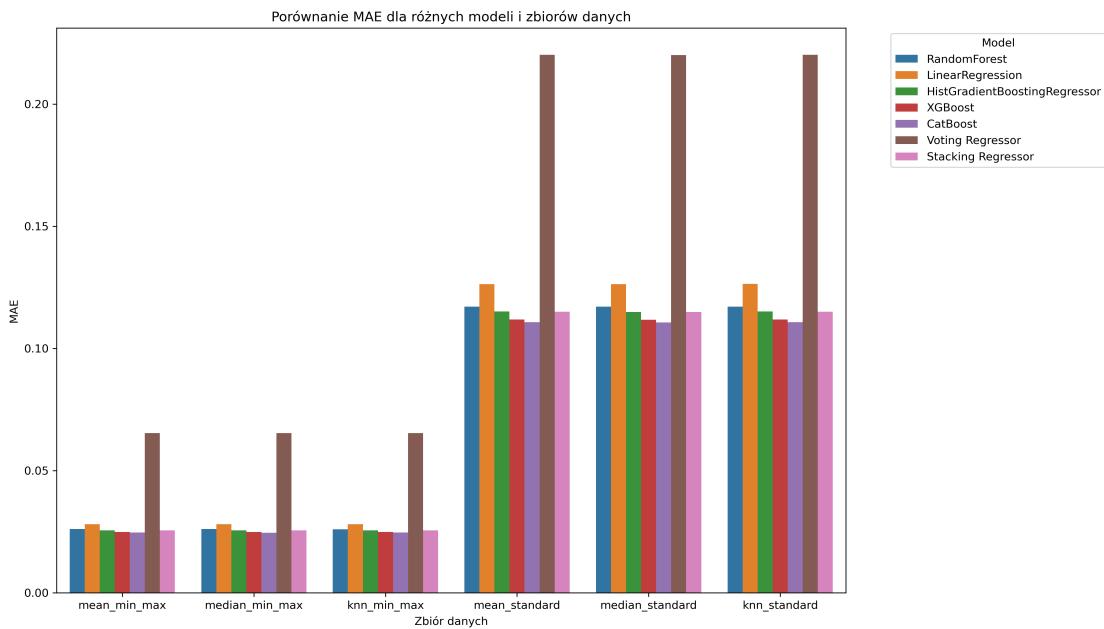


Figure 19: Porównanie MAE dla różnych modeli i zbiorów danych.

Na wykresie (Rys. 19) przedstawiono wartości MAE dla wszystkich modeli na sześciu zbiorach danych. Modele **CatBoost** i **XGBoost** osiągnęły najlepsze wyniki, szczególnie na zbiorach przetworzonych za pomocą `min-max scaling`. **Voting Regressor** uzyskała najwyższe wartości MAE, co wskazuje na jej niższą skuteczność w porównaniu do innych modeli.

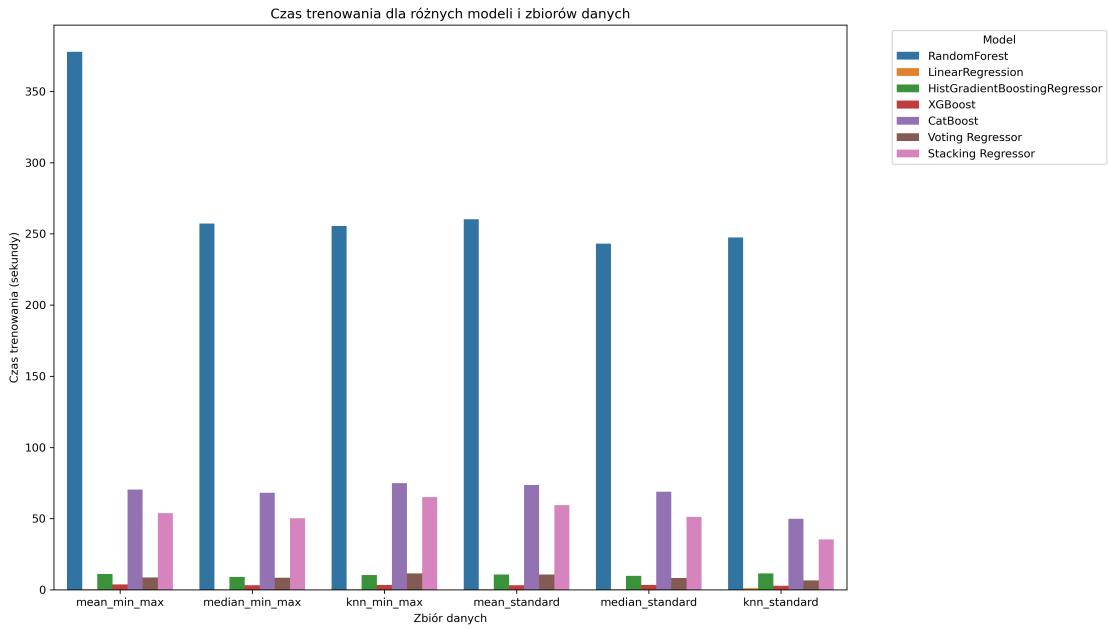


Figure 20: Czas trenowania dla różnych modeli i zbiorów danych.

Na wykresie (Rys. 20) zaprezentowano czas trenowania poszczególnych modeli na różnych zbiorach danych. Modele **Linear Regression** i **XGBoost** charakteryzowały się najkrótszym czasem trenowania (zaledwie kilka sekund na zbiór danych), co czyni je wyjątkowo wydajnymi. Natomiast **Random Forest** wymagał znacznie więcej czasu obliczeniowego, co może stanowić problem w przypadku dużych zbiorów danych. Modele uczenia zespołowego także wymagały więcej czasu niż modele podstawowe.

9.2.6 Walidacja krzyżowa

W walidacji krzyżowej nie został użyty model Random Forest za względu na bardzo długi czas obliczeń.

Table 3: Wyniki cross-walidacji.

ID	Dataset	Model	Manual MAE	Sklearn MAE	Manual RMSE	Sklearn RMSE	R^2
0	mean_min_max	LinearRegression	0.028125	0.028125	0.042104	0.042104	0.964139
1	mean_min_max	XGBoost	0.024906	0.024907	0.037639	0.037616	0.971340
2	mean_min_max	CatBoost	0.024669	0.024676	0.037222	0.037232	0.971972
3	mean_min_max	HistGradientBoosting	0.025667	0.025650	0.038547	0.038553	0.969942
4	mean_min_max	Voting Regressor	0.065545	0.065536	0.083119	0.083114	0.860243
5	mean_min_max	Stacking Regressor	0.025653	0.025713	0.038502	0.038597	0.970012
6	median_min_max	LinearRegression	0.028125	0.028125	0.042104	0.042104	0.964139
7	median_min_max	XGBoost	0.024905	0.024896	0.037649	0.037621	0.971326
8	median_min_max	CatBoost	0.024665	0.024675	0.037220	0.037219	0.971975
9	median_min_max	HistGradientBoosting	0.025663	0.025659	0.038550	0.038560	0.969937
10	median_min_max	Voting Regressor	0.065539	0.065532	0.083112	0.083109	0.860266
11	median_min_max	Stacking Regressor	0.025650	0.025727	0.038506	0.038606	0.970005
12	knn_min_max	LinearRegression	0.028148	0.028148	0.042127	0.042127	0.964100
13	knn_min_max	XGBoost	0.024895	0.024905	0.037604	0.037613	0.971394
14	knn_min_max	CatBoost	0.024681	0.024678	0.037226	0.037219	0.971966
15	knn_min_max	HistGradientBoosting	0.025663	0.025663	0.038544	0.038555	0.969947
16	knn_min_max	Voting Regressor	0.065494	0.065490	0.083064	0.083063	0.860425
17	knn_min_max	Stacking Regressor	0.025647	0.025705	0.038498	0.038563	0.970019
18	mean_standard	LinearRegression	0.126496	0.126496	0.189368	0.189368	0.964139
19	mean_standard	XGBoost	0.111978	0.112011	0.169199	0.169189	0.971371
20	mean_standard	CatBoost	0.110914	0.111003	0.167338	0.167466	0.971996
21	mean_standard	HistGradientBoosting	0.115442	0.115366	0.173369	0.173398	0.969942
22	mean_standard	Voting Regressor	0.220838	0.220809	0.287181	0.287173	0.917526
23	mean_standard	Stacking Regressor	0.115399	0.115741	0.173163	0.173536	0.970014
24	median_standard	LinearRegression	0.126496	0.126496	0.189368	0.189368	0.964139
25	median_standard	XGBoost	0.112033	0.112007	0.169323	0.169222	0.971329
26	median_standard	CatBoost	0.110947	0.110959	0.167407	0.167389	0.971973
27	median_standard	HistGradientBoosting	0.115425	0.115407	0.173386	0.173428	0.969937
28	median_standard	Voting Regressor	0.220815	0.220783	0.287153	0.287145	0.917542
29	median_standard	Stacking Regressor	0.115386	0.115801	0.173185	0.173577	0.970006
30	knn_standard	LinearRegression	0.126601	0.126601	0.189472	0.189472	0.964100
31	knn_standard	XGBoost	0.111979	0.112037	0.169150	0.169213	0.971387
32	knn_standard	CatBoost	0.110999	0.110999	0.167440	0.167424	0.971962
33	knn_standard	HistGradientBoosting	0.115423	0.115400	0.173356	0.173384	0.969947
34	knn_standard	Voting Regressor	0.220862	0.220841	0.287228	0.287216	0.917500
35	knn_standard	Stacking Regressor	0.115369	0.115763	0.173144	0.173561	0.970020

9.2.7 Wnioski

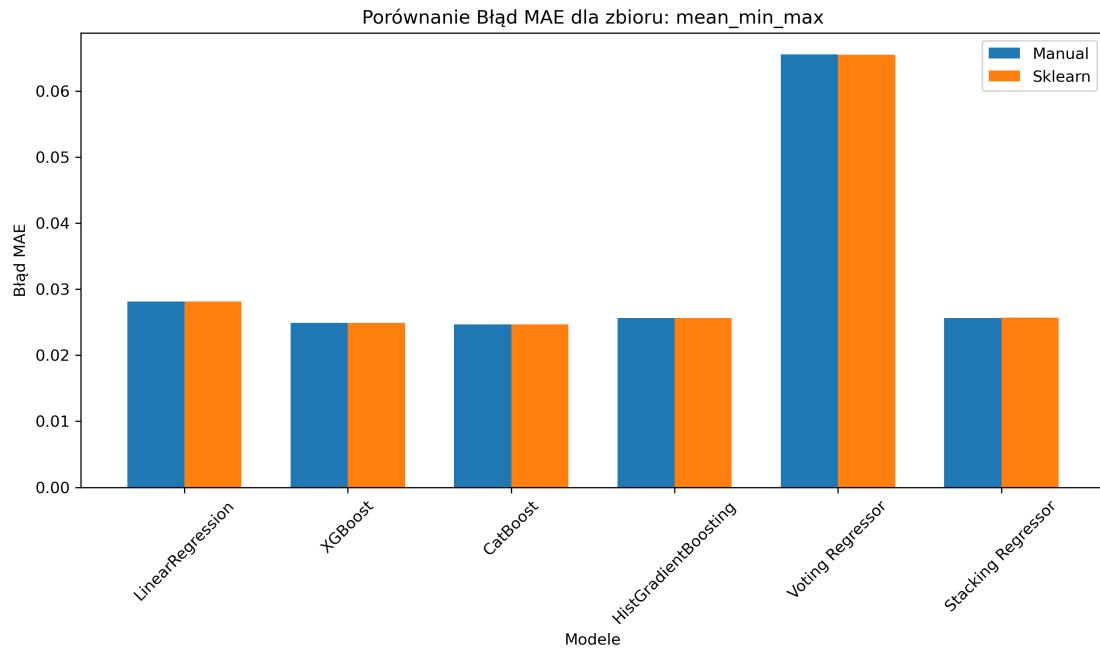


Figure 21: Porównanie błędu MEA dla zbioru mean_max_min.

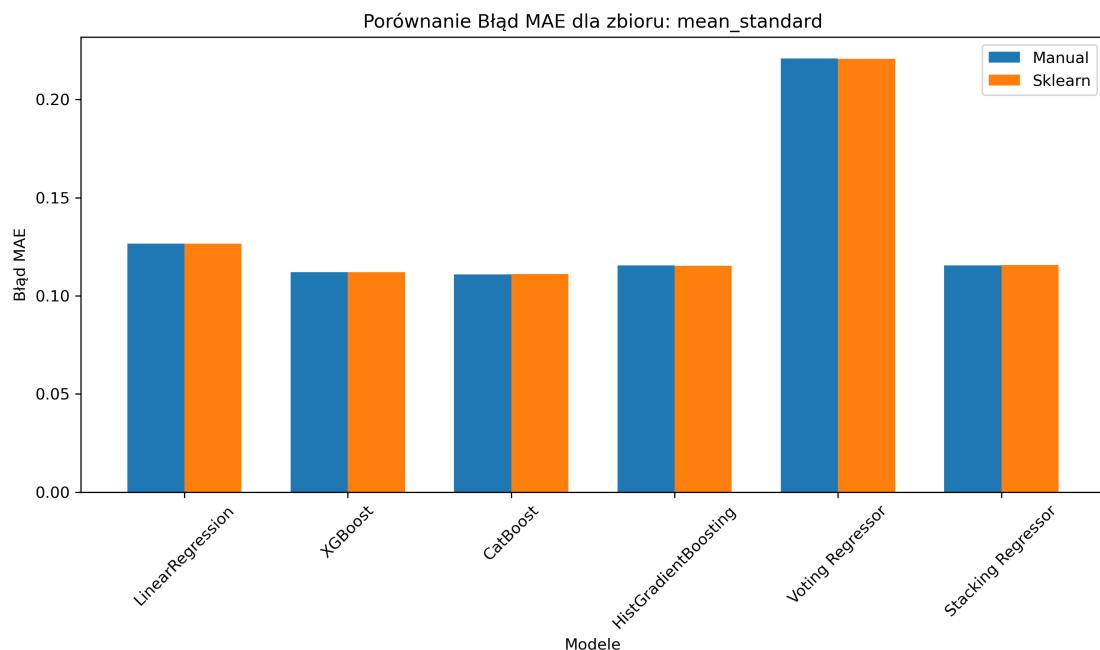


Figure 22: Porównanie błędu MEA dla zbioru mean_standard.

- XGBoost i CatBoost oraz HistGradientBoosting wykazują najlepsze wyniki w MAE, RMSE oraz R^2 , co sugeruje, że są one najbardziej odpowiednie dla analizowanych zbiorów danych.
- Modele z normalizacją (_min_max) działają generalnie lepiej niż te z użyciem standaryzacji (_standard), co może wynikać z charakteru danych.
- Voting Regressor osiąga najsłabsze wyniki w każdym scenariuszu, co może sugerować, że średnia z modeli bazowych nie jest optymalna dla tych danych.
- Różnice między wynikami obliczonymi manualnie i za pomocą Sklearn są minimalne, co potwierdza poprawność implementacji obliczeń manualnych.

9.2.8 Wnioski z analizy

- **CatBoost** i **XGBoost** oraz **HistGradientBoostingRegressor** okazały się najskuteczniejszymi modelami, osiągając zarówno wysoką dokładność predykcji (niski MAE i RMSE), jak i akceptowalny czas trenowania. Wskazuje to na ich przydatność w zadaniach związanych z predykcją cen samochodów.
- Modele **Linear Regression** oraz **Random Forest** miały znaczące ograniczenia - pierwszy charakteryzował się niską dokładnością, a drugi wymagał znacznie dłuższego czasu obliczeniowego, co czyni go mniej praktycznym w zastosowaniach na dużych zbiorach danych.
- Analiza wskazała, że przetwarzanie danych za pomocą **min-max scaling** było bardziej efektywne niż standaryzacja, co podkreśla znaczenie odpowiedniego przygotowania danych w procesie modelowania.

9.3 Optymalizacja

W wyniku otrzymanych rezultatów do optymalizacji zostanie wykorzystany model **Histogram-based Gradient Boosting Regressor** ze względu na czas działania i skuteczność. Wykorzystany zostanie zbiór **mean_min_max.csv**.

9.3.1 Optymalizacja cech

Na tym etapie projektu przeprowadzono optymalizację cech, aby zredukować liczbę zmiennych wejściowych i poprawić efektywność modelowania. Wykorzystano następujące metody:

1. Macierz korelacji

Na podstawie macierzy korelacji:

- Odrzucono cechy o niskiej korelacji. Próg korelacji ustalono na wartość 0.1. Cechy, dla których wartości korelacji bezwzględnej były mniejsze niż 0.1, zostały usunięte.
- Dodatkowo, aby sztucznie obniżyć jakość uzyskanych modeli przed optymalizacją hiperparamterów, usunięto kilka wysoko skorelowanych zmiennych. Był to: `car_age`, `odometer`, `sale_year`, `condition`, `mmr`, `sale_month`.

Końcowo zredukowany zbór cech przedstawiono na Rysunku 23 przedstawiającym macierz korelacji po odrzuceniu nadmiarowych cech.

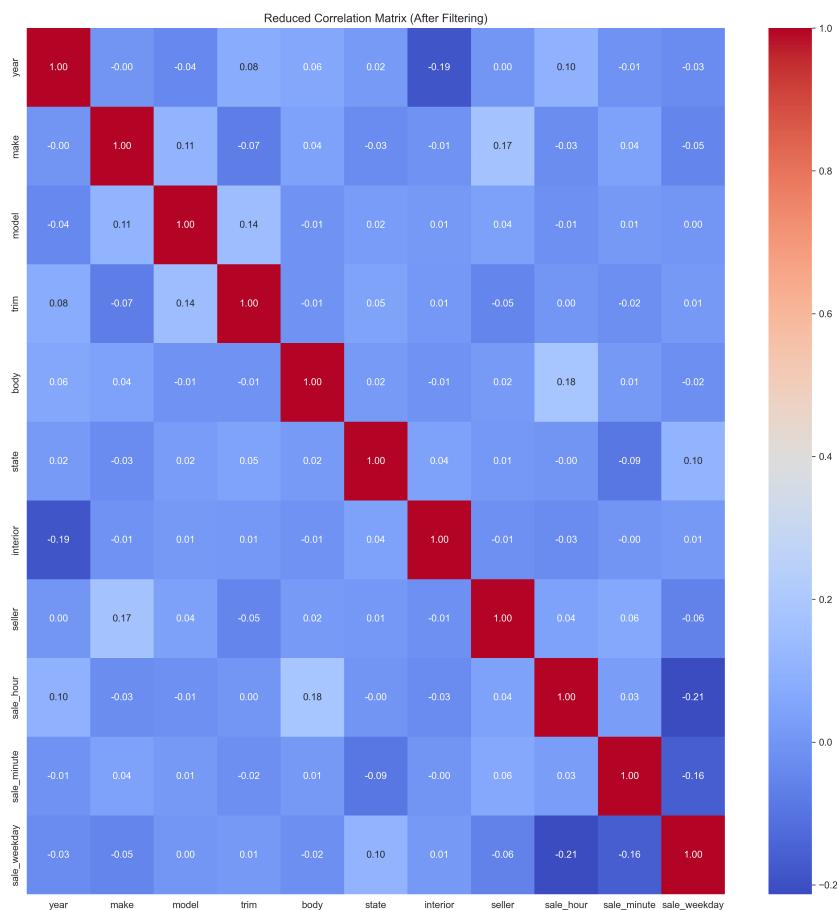


Figure 23: Macierz korelacji po odrzuceniu nadmiarowych cech.

2. Recursive Feature Elimination (RFE)

W celu dalszej optymalizacji cech zastosowano algorytm Recursive Feature Elimination (RFE). Jako model bazowy wybrano regresję liniową (`LinearRegression`). Algorytm iteracyjnie analizował cechy, jednak żadna z nich nie została usunięta, co wskazuje na ich wysoką wartość predykcyjną.

- **Wynik:** Wszystkie cechy zostały zachowane: `year`, `make`, `model`, `trim`, `body`, `state`, `interior`, `seller`, `sale_hour`, `sale_minute`, `sale_weekday`.

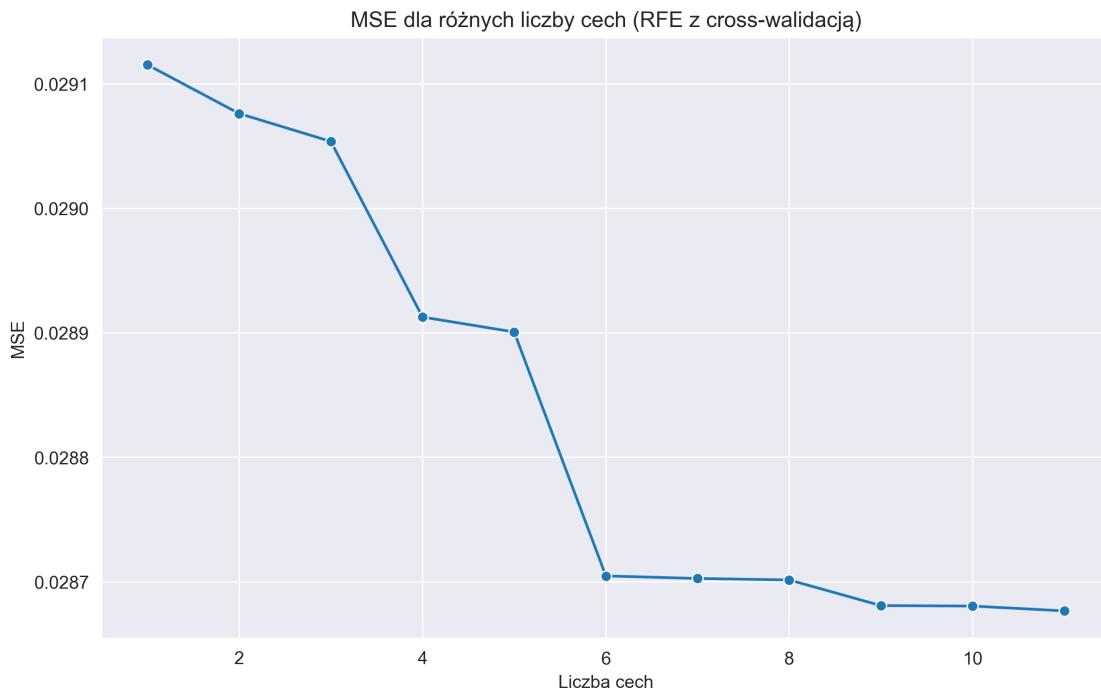


Figure 24: Wykres MSE dla różnych liczb cech (RFE z cross-walidacją).

3. Algorytmy zaawansowane - Particle Swarm Optimization (PSO)

Aby uzyskać ocenę metod bardziej zaawansowanych i osiągnąć potencjalnie wyższą ocenę, wdrożono optymalizację za pomocą Particle Swarm Optimization (PSO). Funkcja celu maksymalizowała skuteczność modelu Histogram-based Gradient Boosting Regressor na podstawie cross-walidacji z jednociesnym uwzględnieniem kary za liczbę cech.

- **Wynik:** Algorytm PSO zidentyfikował jedną kluczową cechę - `year`

4. Genetic Algorithm (GA)

Przeprowadzono optymalizację cech za pomocą Genetic Algorithm (GA). Algorytm poszukiwał optymalnego podzbioru cech, minimalizując błąd średniokwadratowy (MSE) modelu Histogram-based Gradient Boosting Regressor z uwzględnieniem kary za liczbę cech.

- W każdej generacji selekcjonowano najlepsze osobniki metodą turniejową, krzyżowano je jednopunktowo, a następnie wprowadzano mutacje z prawdopodobieństwem 0.1.
- **Wynik:** Wybrane cechy to `year`, `state`, `interior`. Finalne wyniki z kolejnych generacji wskazywały stabilizację funkcji celu, co sugeruje skuteczne poszukiwanie optymalnego zestawu cech.

5. Feature Importance - Extra Trees Regressor

Przeprowadzono ranking ważności cech z wykorzystaniem modelu Extra Trees Regressor. Progiem ważności uznano 0.01, odrzucając cechy mniej istotne. Algorytm analizował wszystkie cechy, jednak żadna z nich nie została usunięta, co wskazuje na ich wartość predykcyjną.

9.3.2 Porównanie metod optymalizacji

Przeprowadzono porównanie wyników z wykorzystaniem wybranych metod optymalizacji na zbiorze walidacyjnym:

Metoda	MAE	MSE	RMSE	R ²
RFE	0.061844	0.007450	0.086312	0.848586
PSO	0.132992	0.028499	0.168818	0.420765
Extra Trees	0.061844	0.007450	0.086312	0.848586
Genetic Algorithm	0.127318	0.020533	0.162891	0.460724

Table 4: Porównanie wyników metod optymalizacji cech.

Najlepszy wynik osiągnęła metoda **RFE**, minimalizując wartości MAE i RMSE, co wskazuje na jej skuteczność w zadaniu optymalizacji cech.

9.3.3 Wyniki optymalizacji cech

Na podstawie przeprowadzonych analiz wybrano zbór cech do dalszego modelowania, ograniczając go do zmiennych:

- year
- make
- model
- trim
- body
- state
- interior
- seller
- sale_hour
- sale_minute
- sale_weekday

9.3.4 Optymalizacja hiperparametrów

W każdej metodzie uwzględniono wartości domyślne dla modelu HistGradientBoosting.

Model Bazowy:

- Mean Squared Error: 0.0070
- R²: 0.8572

GridSearchCV:

- Mean Squared Error (Testowy): 0.0046
- R² (Testowy): 0.9063
- Najlepsze Hiperparametry: {'l2_regularization' : 0.0,'learning_rate' : 0.2,'max_depth' : None,'max_iter' : 300,'min_samples_leaf' : 20}

RandomizedSearchCV:

- Mean Squared Error (Testowy): 0.0046
- R² (Testowy): 0.9061

- Najlepsze Hiperparametry: `{'min_samples_leaf': 50, 'max_iter': 300, 'max_depth': None, 'learning_rate': 0.2, 'l2_regularization': 0.0}`

Optuna:

- Mean Squared Error (Testowy): 0.0042
- R^2 (Testowy): 0.9151
- Najlepsze Hiperparametry: `{'learning_rate': 0.2930895840551921, 'max_iter': 487, 'max_depth': None, 'min_samples_leaf': 20, 'l2_regularization': 3.94153331112236}`

Algorytm Genetyczny:

- Mean Squared Error (Testowy): 0.0043
- R^2 (Testowy): 0.9130
- Najlepsze Hiperparametry: `{'learning_rate': 0.29080400772472587, 'max_iter': 497, 'max_depth': 6, 'min_samples_leaf': 34, 'l2_regularization': 2.30621555056688}`

Z porównania różnych metod optymalizacji hiperparametrów, widać, że najlepsze wyniki uzyskano przy użyciu Optuna. Osiągnięto najwyższy błąd średniokwadratowy (Mean Squared Error) na poziomie 0.0042 oraz najwyższy współczynnik determinacji (R^2) równy 0.9151.

Metody GridSearchCV i RandomizedSearchCV osiągnęły zbliżone wyniki, z wartościami MSE na poziomie 0.0046 i R^2 odpowiednio 0.9063 i 0.9061. Różnice te wskazują na wysoką efektywność obu metod, choć Optuna daje lepsze rezultaty.

Algorytm genetyczny uzyskał wartości MSE na poziomie 0.0043 oraz R^2 równy 0.9130, co również jest bardzo bliskie wynikowi Optuna, choć z nieco gorszymi hiperparametrami. Był także najbardziej czasochłonną metodą - zarówno przy implementacji jak i wykonaniu.

Optuna okazała się najlepszą metodą w kontekście uzyskanych wyników, jednak inne podejścia, takie jak GridSearchCV i RandomizedSearchCV, również dają bardzo dobre rezultaty (a są prostsze w implementacji) i mogą być odpowiednie w różnych sytuacjach zależnych od wymagań czasowych i obliczeniowych.

10 Podsumowanie

W ramach projektu przeprowadzono kompleksową analizę i predykcję cen samochodów używanych na podstawie danych historycznych z aukcji. Główne etapy projektu obejmowały przygotowanie danych, wybór cech, budowę modeli predykcyjnych, optymalizację cech oraz optymalizację hiperparametrów.

Przygotowanie danych

Etap wstępnego przetwarzania danych był kluczowy dla zapewnienia spójności i jakości zbioru danych. Wykonano następujące kroki:

- Obliczono wiek samochodów oraz wyodrębniono dodatkowe cechy (np. godzina, minuta i dzień tygodnia sprzedaży) na podstawie daty sprzedaży.
- Usunięto wartości odstające metodą IQR, co pozwoliło na wyeliminowanie około 8,3% danych.
- Zmieniono zmienne kategoryczne na wartości liczbowe za pomocą kodowania etykiet (label encoding), co umożliwiło skuteczne wykorzystanie modeli predykcyjnych.
- Zastosowano imputację brakujących danych trzema metodami: średnią, medianą oraz k-Nearest Neighbors.
- Przeprowadzono skalowanie i standaryzację danych, tworząc różne zestawy przetworzonych danych (min-max scaling, standard scaling).

Modelowanie i ocena

Do modelowania wykorzystano różnorodne algorytmy regresji, takie jak:

- **CatBoost Regressor**, który osiągnął najlepsze wyniki pod względem MAE, RMSE i współczynnika determinacji R^2 , jednocześnie utrzymując stosunkowo krótki czas trenowania.
- **XGBoost Regressor**, który oferował wyniki porównywalne z CatBoostem, ale z nieco szybszym czasem trenowania.
- Modele takie jak **Random Forest** i **Linear Regression** wykazały swoje ograniczenia — pierwszy wymagał dużego czasu obliczeniowego, a drugi charakteryzował się niską dokładnością.
- **Voting Regressor** okazał się najmniej skuteczny, co wskazuje na ograniczenia metod ensemble w tym przypadku.

Podsumowując wyniki, przetwarzanie danych za pomocą skalowania min-max wykazało większą efektywność niż standaryzacja, co uwypukla znaczenie odpowiedniego przygotowania danych w procesie modelowania.

Optymalizacja cech

W celu redukcji wymiarowości danych zastosowano różnorodne techniki optymalizacji cech:

- Algorytm **Recursive Feature Elimination (RFE)** nie wykluczył żadnej cechy, co wskazuje na wysoką wartość predykcyjną wszystkich zmiennych.
- Algorytm **Extra Trees Regressor** również nie wykluczył żadnej cechy, uzyskując wyniki identyczne do RFE, co wskazuje na spójność między tymi metodami.
- **Particle Swarm Optimization (PSO)** oraz **Genetic Algorithm (GA)** osiągnęły podobne wyniki, identyfikując ograniczoną liczbę kluczowych cech (np. `year`), jednak ich skuteczność była niższa w porównaniu do metod RFE i Extra Trees Regressor.

Optymalizacja cech pozwoliła na zachowanie kluczowych zmiennych bez utraty jakości modelowania.

Optymalizacja hiperparametrów

W ramach optymalizacji hiperparametrów zastosowano cztery różne metody:

- **GridSearchCV** oraz **RandomizedSearchCV**, które osiągnęły porównywalne wyniki (MSE na poziomie 0.0046 i R^2 na poziomie 0.9063).
- **Optuna** okazała się najskuteczniejsza, uzyskując najniższe MSE na poziomie 0.0042 oraz najwyższy współczynnik determinacji R^2 równy 0.9151.
- **Algorytm Genetyczny** dostarczył wyniki zbliżone do Optuny (MSE: 0.0043, R^2 : 0.9130), jednak charakteryzował się większą złożonością obliczeniową.

Optymalizacja hiperparametrów znacząco poprawiła wydajność modeli, a Optuna została uznana za najlepszą metodę w kontekście uzyskanych wyników i efektywności obliczeniowej.

Wnioski i rekommendacje

Wyniki projektu wskazują, że najlepsze modele predykcyjne to CatBoost i XGBoost, które charakteryzują się wysoką dokładnością i krótkim czasem trenowania. Optymalizacja cech, szczególnie za pomocą RFE i Extra Trees Regressor, dostarczyła najbardziej wartościowych wyników, wskazując na istotność wszystkich cech w predykcji. Ostateczny model wykorzystał Histogram-based Gradient Boosting Regressor, który oferuje równowagę między dokładnością a wydajnością obliczeniową.

Projekt dostarczył cennych informacji na temat metod predykcji cen samochodów używanych i możliwości ich zastosowania w praktyce. W przyszłości warto rozważyć integrację bardziej zaawansowanych metod optymalizacji oraz uwzględnienie dynamicznych danych rynkowych dla dalszej poprawy dokładności modeli.

References

- [1] Midoel3ila. Car used priced prediction using ml, 2024. Accessed: 2024-12-20.
- [2] Zabihullah. Car price prediction using ml, 2018. Accessed: 2024-12-20.