Przewidywanie cen sprzedaży domu - model regresyjny

Maja Fiszer (223354), Weronika Kłujszo (223599)

26 stycznia 2026

Spis treści

1 Wprowadzenie

1.1 Cel projektu

Celem projektu jest zbudowanie modelu regresyjnego umożliwiającego przewidywanie ceny sprzedaży domu na podstawie zestawu cech opisujących nieruchomość. Analiza opiera się na zbiorze danych z konkursu House Prices - Advanced Regression Techniques udostępnionym na platformie Kaggle.

1.2 Opis problemu biznesowego i naukowego

Rynek nieruchomości stanowi jeden z kluczowych elementów gospodarki, a dokładna wycena domów ma znaczenie zarówno dla kupujących, sprzedających, jak i instytucji finansowych. Ceny nieruchomości są determinowane przez wiele czynników, takich jak lokalizacja, powierzchnia, stan techniczny, liczba pomieszczeń czy standard wykończenia.

Z naukowego punktu widzenia projekt skupia się na analizie wpływu poszczególnych cech na cenę domu oraz na badaniu efektywności różnych technik uczenia maszynowego w kontekście regresji. Celem jest określenie, które zmienne mają największą wartość prognostyczną oraz jakie metody modelowania najlepiej sprawdzają się w praktyce.

2 Eksploracja i analiza danych

2.1 Opis zbioru danych

Zbiór danych pochodzi z konkursu na stronie Kaggle i zawiera zmienne opisujące nieruchomości oraz jedną zmienną objaśnianą `SalePrice`. Dane obejmują informacje dotyczące:

- działki i jej parametrów,
- konstrukcji dom,
- pomieszczeń i ich jakości,
- garażu, piwnicy oraz innych elementów infrastruktury,
- lokalizacji.

Zbiór treningowy zawiera 1460 obserwacji, natomiast zbiór testowy 1459 obserwacji, przy czym w zbiorze testowym nie podano cen sprzedaży.

2.2 Opis zmiennych

Zbiór danych zawiera 79 zmiennych opisujących cechy nieruchomości, które służą do przewidywania ceny sprzedaży domu. Zostały one podzielone na zmienne ilościowe (inaczej numeryczne) oraz kategoryczne.

- **Ilościowe:**
  - **LotFrontage** – długość frontu działki przylegającej do ulicy (w stopach),
  - **LotArea** – powierzchnia działki w stopach kwadratowych,
  - **MasVnrArea** – powierzchnia zewnętrznej okładziny kamiennej (w stopach kwadratowych),
  - **BsmtFinSF1** – powierzchnia wykończonej części piwnicy typu 1,
  - **BsmtFinSF2** – powierzchnia wykończonej części piwnicy typu 2,
  - **BsmtUnfSF** – powierzchnia niewykończonej części piwnicy,
  - **TotalBsmtSF** – łączna powierzchnia piwnicy,
  - **1stFlrSF** – powierzchnia pierwszej kondygnacji,
  - **2ndFlrSF** – powierzchnia drugiej kondygnacji,
  - **LowQualFinSF** – powierzchnia wykończenia niskiej jakości,
  - **GrLivArea** – powierzchnia mieszkalna nad poziomem gruntu,
  - **GarageArea** – powierzchnia garażu,
  - **WoodDeckSF** – powierzchnia drewnianego tarasu,
  - **OpenPorchSF** – powierzchnia otwartego ganku,
  - **EnclosedPorch** – powierzchnia zabudowanego ganku,
  - **3SsnPorch** – powierzchnia ganku trzysezonowego,
  - **ScreenPorch** – powierzchnia ganku z siatką,
  - **PoolArea** – powierzchnia basenu,
  - **MiscVal** – wartość dodatkowych elementów (np. szop, altan),

  - **YearBuilt** – rok wybudowania domu,
  - **YearRemodAdd** – rok ostatniego remontu,
  - **GarageYrBlt** – rok budowy garażu,
  - **MoSold** – miesiąc sprzedaży (1–12),
  - **YrSold** – rok sprzedaży,
  - **BedroomAbvGr** – liczba sypialni nad gruntem,
  - **KitchenAbvGr** – liczba kuchni nad gruntem,
  - **TotRmsAbvGrd** – całkowita liczba pokoi nad gruntem,
  - **Fireplaces** – liczba kominków,
  - **GarageCars** – pojemność garażu wyrażona w liczbie samochodów,
  - **FullBath** – liczba w pełni wyposażonych łazienek,
  - **HalfBath** – liczba łazienek z toaletą,
  - **BsmtFullBath** – liczba w pełni wyposażonych łazienek w piwnicy,
  - **BsmtHalfBath** – liczba łazienek w piwnicy,
  - **OverallQual** – ogólna jakość materiałów i wykończenia domu (1–10),
  - **OverallCond** – ogólny stan techniczny domu (1–10),
  - **MSSubClass** – typ obiektu budowlanego (klasa zabudowy),
  - **SalePrice** – cena sprzedaży domu (zmienna objaśniana).

- **Kategoryczne:**
  - **MSZoning** – klasyfikacja strefy zagospodarowania przestrzennego,
  - **Street** – typ dostępu do działki (np. nawierzchnia),
  - **Alley** – typ dostępu do działki od tylnej alejki,
  - **LotShape** – kształt działki,
  - **LandContour** – konfiguracja terenu,
  - **Utilities** – dostępne media komunalne,

– **LotConfig** – konfiguracja działki względem ulicy i sąsiadów,
– **LandSlope** – nachylenie działki,
– **Neighborhood** – nazwa dzielnicy,
– **Condition1** – stan otoczenia (np. bliskość dróg, torów),
– **Condition2** – drugi stan otoczenia (jeśli występuje),
– **BldgType** – typ budynku (np. wolnostojący, bliźniak),
– **HouseStyle** – styl architektoniczny budynku,
– **RoofStyle** – typ dachu,
– **RoofMatl** – materiał pokrycia dachowego,
– **Exterior1st** – materiał zewnętrzny (pierwszy),
– **Exterior2nd** – materiał zewnętrzny (drugi),
– **MasVnrType** – typ okładziny murarskiej,
– **ExterQual** – jakość materiałów zewnętrznych,
– **ExterCond** – stan techniczny materiałów zewnętrznych,
– **Foundation** – rodzaj fundamentu,
– **BsmtQual** – jakość piwnicy,
– **BsmtCond** – stan techniczny piwnicy,
– **BsmtExposure** – poziom odsłonięcia piwnicy (okna),
– **BsmtFinType1** – typ wykończenia piwnicy (sekcja 1),
– **BsmtFinType2** - typ wykończenia piwnicy (sekcja 2),
– **Heating** - typ ogrzewania,
– **HeatingQC** – jakość instalacji grzewczej,
– **CentralAir** – obecność klimatyzacji,
– **Electrical** – typ instalacji elektrycznej,
– **KitchenQual** – jakość kuchni,
– **Functional** – ogólna funkcjonalność domu,
– **FireplaceQu** – jakość kominka,
– **GarageType** – typ garażu,
– **GarageFinish** – poziom wykończenia garażu,

- **GarageQual** – jakość garażu,
- **GarageCond** – stan techniczny garażu,
- **PavedDrive** – typ podjazdu,
- **PoolQC** – jakość basenu,
- **Fence** – typ ogrodzenia,
- **MiscFeature** – dodatkowe elementy (np. szopa),
- **SaleType** – typ sprzedaży,
- **SaleCondition** – warunki sprzedaży.

2.3 Analiza brakujących wartości

W zbiorze danych występuje wiele braków, szczególnie w zmiennych opisujących rzadziej spotykane elementy wyposażenia:

- `PoolQC` (jedynie 7 niepustych rekordów), `MiscFeature` (54 znanych wartości), `Alley` (91), `Fence` (281) mają liczne braki wynikające z braku tych elementów w większości domów - te zmienne nie będą przydatne w trakcie analizy,
- zmienne związane z piwnicą i garażem (czyli z przedrostkami `Bsmt` i `Garage`) zawierają braki, gdy dom nie posiada odpowiednio piwnicy lub garażu,
- zmienne takie jak `LotFrontage`, `MasVnrType` (tutaj przykładowo znamy około $\frac{1}{3}$ wszystkich rekordów) oraz `FireplaceQu` mają zauważalną, ale nie drastyczną liczbę braków, co sugeruje konieczność imputacji, na przykład medianą w obrębie danej dzielnicy,

2.4 Analiza wartości odstających

Poniżej przedstawiono zmienne, w których występują wartości odstające oraz opis potencjalnych przyczyn.

Powierzchnia mieszkalna i piwnice:

- `GrLivArea` — bardzo duże powierzchnie mieszkalne (np. luksusowe domy) znacząco odbiegają od średniej,

- `TotalBsmtSF`, `BsmtFinSF1`, `BsmtFinSF2` — niektóre domy nie mają piwnicy, co powoduje dużą liczbę zer, a kilka domów ma bardzo duże piwnice; wartości odstające są naturalne, ale widoczne na wykresach, na przykład boxplotach.

Powierzchnia działki:

- `LotArea`, `LotFrontage` — występują działki o nietypowo dużej powierzchni.

Inne powierzchnie i udogodnienia:

- `1stFlrSF`, `LowQualFinSF`, `GarageArea`, `WoodDeckSF`, `OpenPorchSF`, `EnclosedPorch`, `3SsnPorch`, `ScreenPorch`, `PoolArea`, `MiscVal` — wiele zer wynika z faktu, że dom nie posiada danej cechy (chociażby brak garażu lub tarasu), przez co średnia oscyluje blisko zera i wszelkie wartości niezerowe bardzo widocznie od niej odbiegają.

Ceny i rok budowy:

- `SalePrice` — rozkład silnie prawostronnie skośny, kilka bardzo drogich domów tworzy wartości odstające,
- `YearBuilt` — najstarsze i najnowsze domy mogą wydawać się odległe od reszty danych,
- `MasVnrArea` — kilka domów z bardzo dużymi lub brakującymi okładzinami z kamienia; wartość 0 występuje w wielu przypadkach, co tworzy odstające obserwacje w porównaniu ze średnią.

2.5 Analiza statystyczna

Poniżej przestawiona jest tabelka z analizą statystyczną surowych, niewyczyszczonych zmiennych:

	count	mean	std	min	Pierwszy kwartyl	Drugi kwartyl	Trzeci kwartyl	max
Id	1460.00	730.50	421.61	1.00	365.75	730.50	1095.25	1460.00
MSSubClass	1460.00	56.90	42.30	20.00	20.00	50.00	70.00	190.00
LotFrontage	1201.00	70.05	24.28	21.00	59.00	69.00	80.00	313.00
LotArea	1460.00	10516.83	9981.26	1300.00	7553.50	9478.50	11601.50	215245.00
OverallQual	1460.00	6.10	1.38	1.00	5.00	6.00	7.00	10.00
OverallCond	1460.00	5.58	1.11	1.00	5.00	5.00	6.00	9.00
YearBuilt	1460.00	1971.27	30.20	1872.00	1954.00	1973.00	2000.00	2010.00
YearRemodAdd	1460.00	1984.87	20.65	1950.00	1967.00	1994.00	2004.00	2010.00
MasVnrArea	1452.00	103.69	181.07	0.00	0.00	0.00	166.00	1600.00
BsmtFinSF1	1460.00	443.64	456.10	0.00	0.00	383.50	712.25	5644.00
BsmtFinSF2	1460.00	46.55	161.32	0.00	0.00	0.00	0.00	1474.00
BsmtUnfSF	1460.00	567.24	441.87	0.00	223.00	477.50	808.00	2336.00
TotalBsmtSF	1460.00	1057.43	438.71	0.00	795.75	991.50	1298.25	6110.00
1stFlrSF	1460.00	1162.63	386.59	334.00	882.00	1087.00	1391.25	4692.00
2ndFlrSF	1460.00	346.99	436.53	0.00	0.00	0.00	728.00	2065.00
LowQualFinSF	1460.00	5.84	48.62	0.00	0.00	0.00	0.00	572.00
GrLivArea	1460.00	1515.46	525.48	334.00	1129.50	1464.00	1776.75	5642.00
BsmtFullBath	1460.00	0.43	0.52	0.00	0.00	0.00	1.00	3.00
BsmtHalfBath	1460.00	0.06	0.24	0.00	0.00	0.00	0.00	2.00
FullBath	1460.00	1.57	0.55	0.00	1.00	2.00	2.00	3.00
HalfBath	1460.00	0.38	0.50	0.00	0.00	0.00	1.00	2.00
BedroomAbvGr	1460.00	2.87	0.82	0.00	2.00	3.00	3.00	8.00
KitchenAbvGr	1460.00	1.05	0.22	0.00	1.00	1.00	1.00	3.00
TotRmsAbvGrd	1460.00	6.52	1.63	2.00	5.00	6.00	7.00	14.00
Fireplaces	1460.00	0.61	0.64	0.00	0.00	1.00	1.00	3.00
GarageYrBlt	1379.00	1978.51	24.69	1900.00	1961.00	1980.00	2002.00	2010.00
GarageCars	1460.00	1.77	0.75	0.00	1.00	2.00	2.00	4.00
GarageArea	1460.00	472.98	213.80	0.00	334.50	480.00	576.00	1418.00
WoodDeckSF	1460.00	94.24	125.34	0.00	0.00	0.00	168.00	857.00
OpenPorchSF	1460.00	46.66	66.26	0.00	0.00	25.00	68.00	547.00
EnclosedPorch	1460.00	21.95	61.12	0.00	0.00	0.00	0.00	552.00
3SsnPorch	1460.00	3.41	29.32	0.00	0.00	0.00	0.00	508.00
ScreenPorch	1460.00	15.06	55.76	0.00	0.00	0.00	0.00	480.00
PoolArea	1460.00	2.76	40.18	0.00	0.00	0.00	0.00	738.00
MiscVal	1460.00	43.49	496.12	0.00	0.00	0.00	0.00	15500.00
MoSold	1460.00	6.32	2.70	1.00	5.00	6.00	8.00	12.00
YrSold	1460.00	2007.82	1.33	2006.00	2007.00	2008.00	2009.00	2010.00
SalePrice	1460.00	180921.20	79442.50	34900.00	129975.00	163000.00	214000.00	755000.00

Tabela 1: Statystyki opisowe zmiennych ilościowych

2.6 Identyfikacja kluczowych zależności

W pierwszym etapie analizy przeprowadzono identyfikację zmiennych najmocniej powiązanych ze zmienną objaśnianą *SalePrice*. Dla zmiennych numerycznych obliczono współczynniki korelacji Pearsona. Wyniki wskazują, że najwyżej skorelowane z ceną są:

- **OverallQual** (0.79) – ogólna jakość wykończenia domu,
- **GrLivArea** (0.71) – powierzchnia użytkowa,
- **GarageCars** (0.64) i **GarageArea** (0.62) – wielkość garażu,
- **TotalBsmtSF** (0.61) – powierzchnia piwnicy,

- **1stFlrSF** (0.61) – powierzchnia pierwszego piętra,
- **FullBath** (0.56) – liczba pełnych łazienek.

Są to kluczowe cechy, które z dużym prawdopodobieństwem będą miały istotne znaczenie w modelach predykcyjnych.

Dla zmiennych kategorycznych obliczono różnicę pomiędzy maksymalną a minimalną średnią ceny w poszczególnych kategoriach. Pozwoliło to określić, które cechy są najbardziej „rozróżniające" pod względem poziomu cen. Najbardziej wpływowe cechy kategoryczne to:

- **Neighborhood** (58,5 tys. USD),
- **ExterQual** (54,4 tys. USD),
- **KitchenQual** (53,7 tys. USD),
- **BsmtQual** (53,5 tys. USD),
- **GarageFinish** (40,8 tys. USD),
- **Foundation**, **GarageType**, **HeatingQC**.

Wyniki te wskazują, że zarówno czynniki strukturalne (jakość kuchni, piwnicy, wykończenia zewnętrznego), jak i lokalizacja mają bardzo duży wpływ na ostateczną cenę nieruchomości.

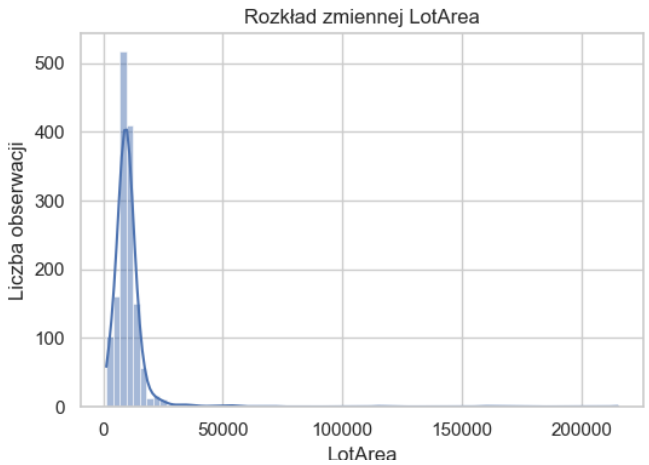
3 Wizualizacja danych

3.1 Wizualizacje rozkładów wszystkich zmiennych

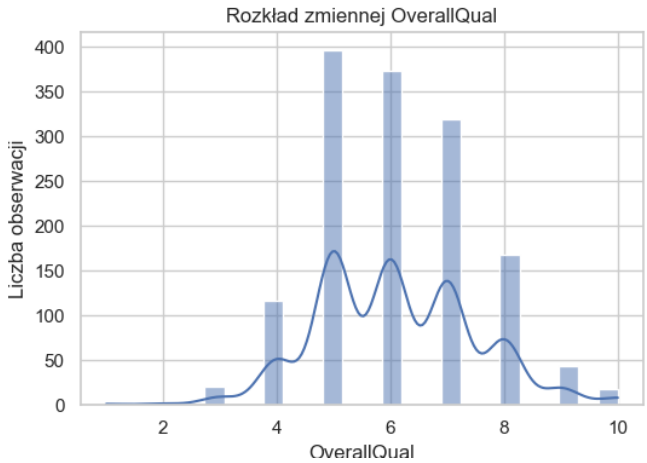
W celu oceny charakteru danych, analizie poddano rozkłady wszystkich zmiennych numerycznych. Zauważalne jest, że wiele zmiennych związanych z **wielkością lub powierzchnią** (m.in. *GrLivArea*, *LotArea*, *TotalBsmtSF*) charakteryzuje się rozkładami **silnie skośnymi** (LotArea: skośność ≈ 12.2) i **wysoce spłaszczonymi** (kurtoza ≈ 202.5). Wskazuje to na dużą koncentrację obserwacji przy niższych wartościach oraz obecność pojedynczych, wyraźnych obserwacji skrajnych (ang. *outliers*). Z kolei zmienne opisujące cechy jakościowe w formie rang liczbowych (np. *OverallQual*, *OverallCond*)

wykazują rozkłady **wielomodalne**, co odzwierciedla strukturę standardów budownictwa wśród analizowanych nieruchomości.

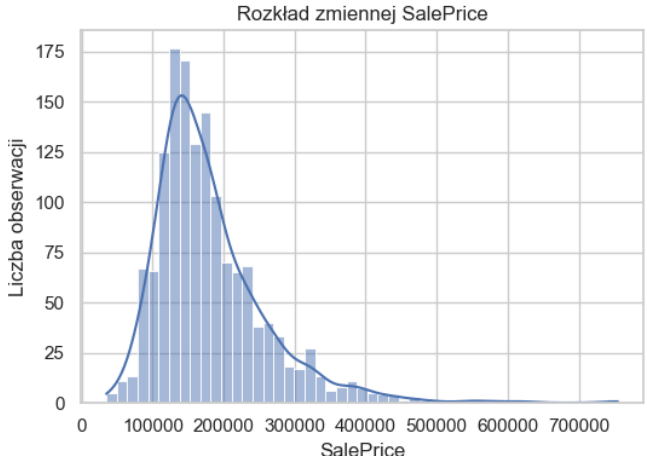
Poniższe wykresy prezentują wybrane przykłady, które ilustrują najbardziej charakterystyczne i zróżnicowane kształty rozkładów w zbiorze danych:



Rys. 1. Rozkład zmiennej *LotArea* (powierzchnia działki). Wyraźna skośność prawostronna, wskazująca na dużą koncentrację małych działek i obecność nielicznych, bardzo dużych działek.



Rys. 2. Rozkład zmiennej *OverallQual* (ogólna jakość). Rozkład wielomodalny z koncentracją na poziomach 5, 6 i 7, co sugeruje, że są to najczęściej spotykane standardy wykończenia.

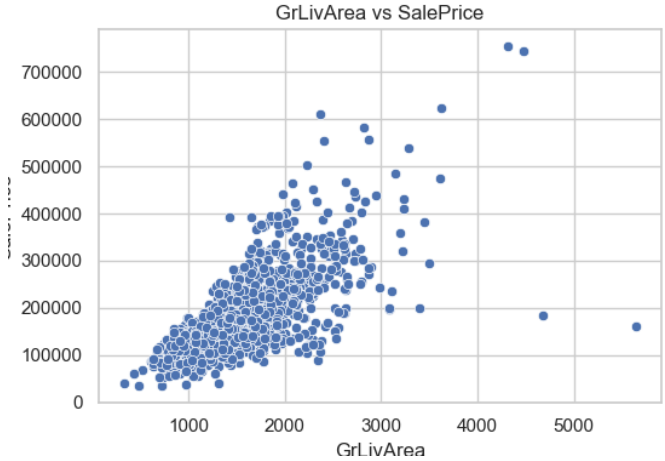


Rys. 3. Rozkład zmiennej *SalePrice* (cena sprzedaży). Dodatnia skośność (≈ 1.88) i długa prawostronna krawędź. Wymagana transformacja logarytmiczna do modelowania.

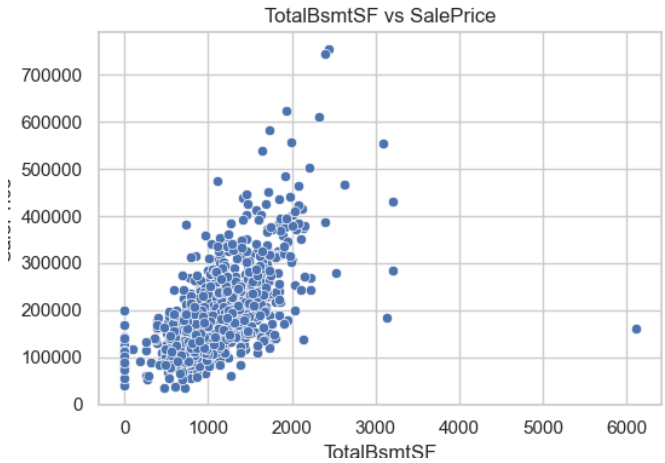
3.2 Wykresy zależności między zmiennymi objaśniającymi a docelowymi

Dla każdej zmiennej numerycznej wygenerowano wykres zależności od ceny sprzedaży (*SalePrice*) w postaci wykresu rozrzutu (*scatterplot*).

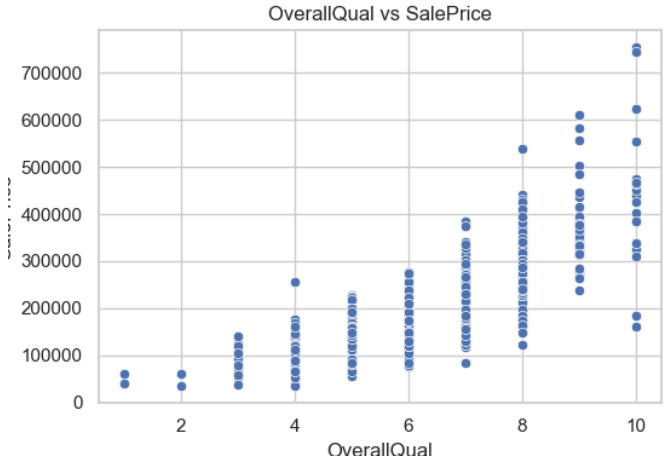
Najsilniejsze relacje pozytywne, wskazujące na bezpośredni związek ze wzrostem ceny, obserwowane są dla: ***GrLivArea*** (powierzchnia mieszkalna), ***TotalBsmtSF*** (całkowita powierzchnia piwnicy), ***OverallQual*** (ogólna jakość) oraz powierzchni poszczególnych kondygnacji.



Rys. 4. Zależność *SalePrice* od *GrLivArea*. Obserwacja silnej, pozytywnej relacji oraz wzrostu wariancji błędu (heteroscedastyczność).



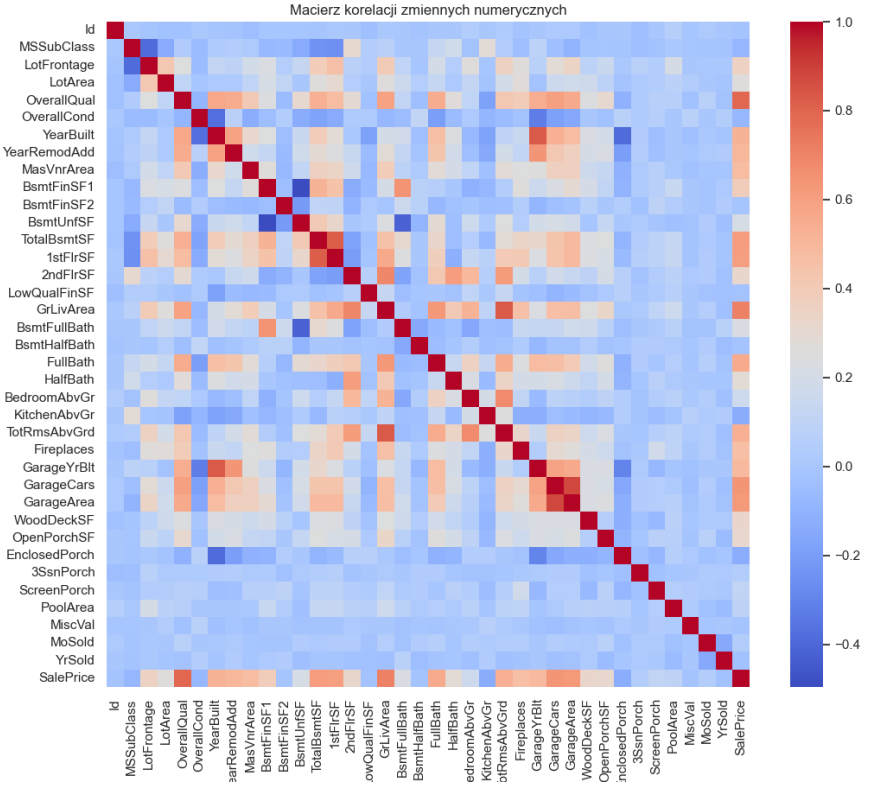
Rys. 5. Zależność *SalePrice* od *TotalBsmtSF*. Relacja jest silna, ale z widoczną koncentracją punktów przy niższych wartościach, co wskazuje na skośny rozkład powierzchni piwnic.



Rys. 6. Zależność *SalePrice* od *OverallQual*. Wyraźna zależność monotoniczna z dużymi skokami mediany cen, co potwierdza, że jest to kluczowy predyktor.

3.3 Macierz korelacji dla zmiennych ciągłych

Aby syntetycznie przedstawić zależności pomiędzy wszystkimi zmiennymi numerycznymi, wygenerowano macierz korelacji. Na wykresie (Rys. 7) widoczne są wyraźne grupy silnie powiązanych cech, m.in. cechy powierzchniowe oraz cechy związane z garażem.

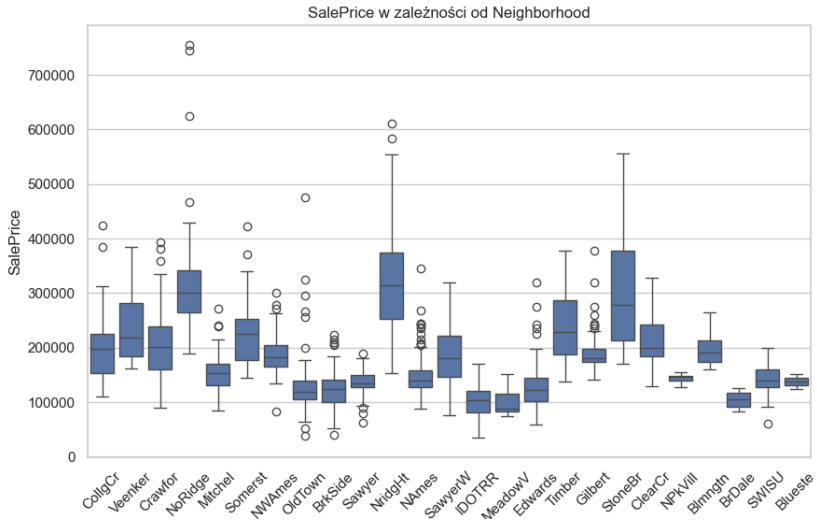


Rys. 7. Macierz korelacji zmiennych numerycznych.

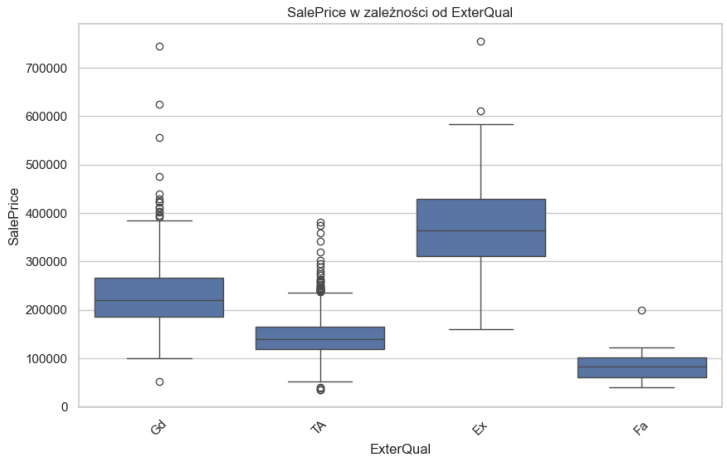
3.4 Wykresy pudełkowe dla analizy zmiennych kategorycznych

Dla każdej zmiennej kategorycznej wygenerowano wykres pudełkowy przedstawiający zróżnicowanie ceny w zależności od kategorii. W wielu przypad-

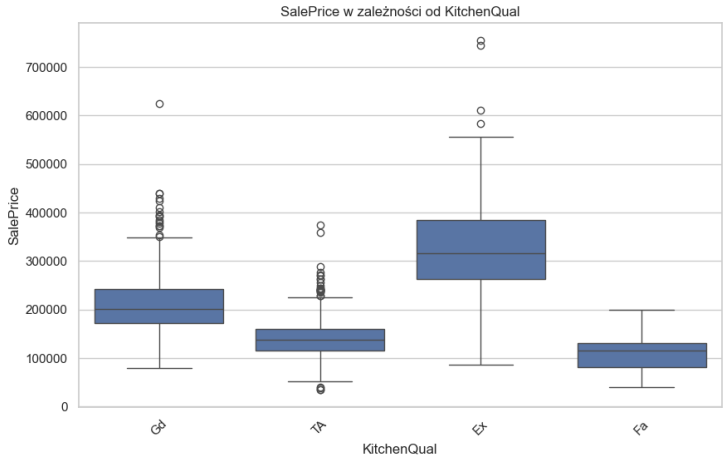
kach — szczególnie dla *Neighborhood* (sąsiedztwo), *ExterQual* (jakość na zewnątrz), *KitchenQual* (jakość kuchni) — widoczne są znaczące różnice pomiędzy medianami cen, potwierdzając wcześniejsze wnioski z analizy statystycznej.



Rys. 8. Wpływ zmiennej *Neighborhood* (Okolica) na rozkład ceny *SalePrice*. Widoczne są drastyczne różnice w medianach cen, co czyni tę zmienną kluczowym predyktorem.



Rys. 9. Wpływ zmiennej *ExterQual* (Jakość materiałów zewnętrznych) na rozkład ceny *SalePrice*. Istnieje wyraźna, monotoniczna relacja między jakością a medianą ceny.



Rys. 10. Wpływ zmiennej *KitchenQual* (Jakość kuchni) na rozkład ceny *SalePrice*. Podobnie jak w przypadku innych cech jakościowych, mediana rośnie wraz ze wzrostem oceny, z minimalnym rozrzutem dla najwyższych kategorii.

3.5 Podsumowanie i wnioski z analizy wizualnej

Przeprowadzona analiza wizualna danych (EDA) oraz wstępna analiza statystyczna pozwoliły na identyfikację kluczowych charakterystyk zbioru danych, które decydują o dalszej strategii modelowania:

- **Wymagana Transformacja Ceny:** Rozkład zmiennej *SalePrice* jest silnie dodatnio skośny (≈ 1.88). Obserwowany na wykresach rozrzutu rosnący rozrzut cen dla droższych nieruchomości (heteroscedastyczność) jednoznacznie wskazuje na konieczność zastosowania *transformacji logarytmicznej* na zmiennej docelowej.
- **Kluczowe Predyktory:** Najsilniejszymi predyktorami liniowymi są: *OverallQual* (0.79) oraz *GrLivArea* (0.71). Wśród zmiennych kategorycznych, największe zróżnicowanie cen generują *Neighborhood* oraz cechy jakościowe (*ExterQual*, *KitchenQual*).
- **Problem Multikolinearności:** Macierz korelacji ujawniła silną zależność wewnętrzną między zmiennymi objaśniającymi, np. *GarageCars* i *GarageArea* oraz *TotalBsmtSF* i *1stFlrSF*. Wymagane jest usunięcie jednej zmiennej z każdej silnie skorelowanej pary.
- **Obserwacje Skrajne:** Wizualizacje rozkładów i wykresy rozrzutu potwierdziły obecność wpływowych obserwacji skrajnych (*outliers*), które mogą destabilizować model i muszą zostać odpowiednio potraktowane w etapie przygotowania danych.

Wnioski te stanowią solidną podstawę do rozpoczęcia etapu Czyszczenia danych.

4 Czyszczenie danych

Dane dotyczące cen domów zostały poddane wstępnemu czyszczeniu i przygotowaniu przed analizą. Proces ten obejmował kilka etapów:

- **Wczytywanie danych** Dane treningowe i testowe zostały wczytane z plików CSV:

  `train.csv`
  `test.csv`

- **Usuwanie kolumn z brakującymi danymi** Kolumny zawierające więcej niż jedną trzecią brakujących wartości zostały usunięte z obu zestawów danych:
  - Funkcja: `remove_missing(train, test)`
- **Uzupełnianie brakujących danych** Pozostałe brakujące wartości zostały uzupełnione:
  - Dla kolumn numerycznych użyto mediany.
  - Dla kolumn kategorycznych użyto wartości modalnej (trybu).
  - Funkcja: `fill_in_missing(train, test)`
- **Analiza i usuwanie wartości odstających (outlierów)** Outliery zostały zidentyfikowane na podstawie IQR (Interquartile Range) dla kolumn numerycznych:
  - Wartości odstające w mniej niż 5% przypadków zostały zastąpione wartościami `NaN`, a następnie usunięte.
  - Wartości odstające w większej liczbie przypadków zostały obcięte do granic `Q1 - 1.5*IQR` i `Q3 + 1.5*IQR`. Zaliczono do nich również około 8% wartości z kolumny *OverallCond*, ale postanowiono ich nie ograniczać.
  - Funkcje: `analyze_outliers(df)`, `remove_outliers(df)`
- **Podsumowanie** Po czyszczeniu dane treningowe miały kształt `(1100, 75)` - podczas pozbywania się outlierów usunięto 360 wierszy, zaś dane testowe `(1459, 74)`. Wszelkie brakujące wartości zostały uzupełnione, a outliery odpowiednio przetworzone lub usunięte, co przygotowało dane do dalszej analizy i modelowania.

4.1 Kodowanie zmiennych kategorycznych

Zmienne kategoryczne zostały zakodowane przy użyciu `OneHotEncoder` z biblioteki scikit-learn. Zastosowano strategię `drop='first'`, która usuwa pierwszą kategorię z każdej zmiennej, aby uniknąć problemu współliniowości, który mógłby być problematyczny w przypadku używania modeli liniowych. Dodatkowo ustawiono parametr `handle_unknown='ignore'`, co po-

zwala modelowi obsługiwać kategorie niewidziane w zbiorze treningowym poprzez utworzenie wektora zerowego dla nieznanych kategorii.

Przed kodowaniem zmienne kategoryczne przechodzą przez `SimpleImputer` ze strategią `most_frequent`, który uzupełnia brakujące wartości najczęściej występującą kategorią. Jest to działanie wykonywane wyłącznie w celu upewnienia się, że dane na pewno wszędzie są uzupełnione. Oryginalnie tym problemem zajmuje się plik `data.py`. Cały proces jest zintegrowany w `ColumnTransformer`, który automatycznie identyfikuje zmienne kategoryczne (typu `object`) i stosuje do nich odpowiedni pipeline.

4.2 Skalowanie i normalizacja

Zmienne numeryczne są skalowane przy użyciu `StandardScaler`, który przekształca cechy do postaci o średniej równej 0 i odchyleniu standardowym równym 1. Jest to szczególnie istotne dla modeli liniowych, takich jak ElasticNet, które są wrażliwe na skalę zmiennych.

Przed skalowaniem zmienne numeryczne przechodzą przez `SimpleImputer` ze strategią `median`, który uzupełnia brakujące wartości medianą danej cechy. Pipeline dla zmiennych numerycznych wygląda następująco:

```
num_pipeline = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])
```

4.3 Feature engineering

Przeprowadzono kompleksowy proces inżynierii cech, który obejmuje następujące transformacje:

Transformacje logarytmiczne: Zastosowano transformację logarytmiczną `np.log1p()` dla zmiennych o skośnym rozkładzie: `LotArea`, `LotFrontage`, `MasVnrArea` i `WoodDeckSF`. Transformacja ta redukuje wpływ wartości odstających i normalizuje rozkład.

Cechy binarne: Utworzono zmienne binarne `HasMasVnrArea` i `HasWoodDeckSF`, które wskazują obecność lub brak danej cechy (wartość większa od 0). Takie podejście pozwala modelowi wychwycić nieliniową zależność między obecnością cechy a ceną.

Cechy czasowe:

- `GarageAge` – wiek garażu obliczony jako różnica między rokiem obecnym a rokiem budowy
- `AgeAtSold` – wiek domu w momencie sprzedaży
- `YearsSinceRemod` – liczba lat od ostatniego remontu do sprzedaży
- `HouseAge` – aktualny wiek domu

Cechy cykliczne: Miesiąc sprzedaży (`MoSold`) został przekształcony na reprezentację trygonometryczną:

$$\texttt{MoSold\_sin} = \sin\left(\frac{2\pi \cdot \texttt{MoSold}}{12}\right)$$
$$\texttt{MoSold\_cos} = \cos\left(\frac{2\pi \cdot \texttt{MoSold}}{12}\right)$$

Dodatkowo utworzono zmienną binarną `HighSeasonSell` wskazującą sezon wysokiej sprzedaży (kwiecień–sierpień).

Cechy agregowane:

- `GrAndBsmtArea` – łączna powierzchnia mieszkalna (naziemna + piwnicy)
- `Bathrooms` – łączna liczba łazienek: $\texttt{FullBath} + \texttt{BsmtFullBath} + 0.5 \cdot (\texttt{HalfBath} + \texttt{BsmtHalfBath})$
- `QualCondScore` – wynik jakości: $\texttt{OverallQual} \times \texttt{OverallCond}$

Usuwanie redundantnych cech: Po utworzeniu nowych cech, oryginalne zmienne zostały usunięte, aby uniknąć redundancji i multikolinearności (np. usunięto `YearBuilt`, `YrSold` po utworzeniu `AgeAtSold`).

4.4 Wybór zmiennych

Zastosowano kilka metod selekcji cech:

DropEmptyColumns: Niestandardowy transformer usuwający kolumny z więcej niż 90% braków danych lub zer. Pomaga to zredukować szum i poprawić stabilność modelu.

CorrelationFiltering: Selekcja cech numerycznych na podstawie korelacji z zmienną docelową. Cechy o korelacji poniżej progu (domyślnie 0.01) są usuwane. Metodę zastosowano jedynie dla modelu liniowego,

FeatureImportanceSelector: Niestandardowy selektor dla modelu drzewiastego wykorzystujący pomocniczy model Gradient Boosting do identyfikacji najważniejszych cech. Cechy o ważności poniżej progu są odrzucane.

SelectKBest: Standardowa metoda z scikit-learn wykorzystująca test statystyczny (`f_regression`) do wyboru k najlepszych cech.

4.5 Podział na zbiór treningowy i testowy

Dane wejściowe zostały dostarczone w postaci dwóch oddzielnych plików CSV: `train_postprocessed.csv` (zbiór treningowy) oraz `test_postprocessed.csv` (zbiór testowy), które zostały wstępnie przetworzone w poprzednich etapach projektu (`data.py`).

Zmienna docelowa `SalePrice` (obecna tylko w zbiorze treningowym) została przekształcona logarytmicznie: $y = \log(1 + \texttt{SalePrice})$, co redukuje skośność rozkładu i stabilizuje wariancję reszt modelu.

Zbiór treningowy został następnie podzielony na podzbiór treningowy i walidacyjny w proporcji 80:20 przy użyciu funkcji `train_test_split` z parametrem `random_state=42` dla zapewnienia reprodukowalności:

```
X_train, X_val, y_train, y_val = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Ostatecznie:

- Zbiór treningowy: $n_{train} = 1168$ obserwacji (80% oryginalnego zbioru treningowego),
- Zbiór walidacyjny: $n_{val} = 292$ obserwacje (20% oryginalnego zbioru treningowego),
- Zbiór testowy: $n_{test} = 1459$ obserwacji (z osobnego pliku, bez zmiennej docelowej).

Zbiór walidacyjny służy do oceny wydajności i porównywania różnych konfiguracji pipeline'ów, natomiast zbiór testowy jest używany do ostatecznych predykcji i ewentualnego przesłania wyników do konkursu Kaggle.

5 Budowa modelu

5.1 Wybór i uzasadnienie wybranych modeli

Ze względu na złożoną strukturę zbioru danych (duża liczba cech, współliniowość, nieliniowe zależności oraz obecność interakcji między zmiennymi), w projekcie zastosowano kilka modeli regresyjnych reprezentujących różne podejścia do problemu predykcji. Pozwala to na porównanie ich zachowania oraz ocenę, które mechanizmy modelowania najlepiej odpowiadają charakterowi danych.

5.1.1 Regresja liniowa

Regresja liniowa stanowi punkt odniesienia dla bardziej zaawansowanych metod. Model ten zakłada liniową zależność pomiędzy zmienną objaśnianą a zestawem predyktorów i jest opisany wzorem:

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon,$$

gdzie y oznacza logarytm ceny sprzedaży domu, x_j to zmienne objaśniające, β_j – współczynniki regresji, a ε składnik losowy.

Zastosowanie regresji liniowej pozwala na ocenę, w jakim stopniu zależności liniowe są wystarczające do opisu danych oraz umożliwia interpretację wpływu poszczególnych cech na cenę nieruchomości. Ze względu na silną współliniowość i dużą liczbę predyktorów model ten traktowany jest wyłącznie jako baza porównawcza.

5.1.2 Regresja Ridge

Regresja Ridge rozszerza klasyczny model liniowy o regularyzację typu L2, która ogranicza wartości współczynników regresji:

$$\min_{\beta} \left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right).$$

Dodanie składnika kary zmniejsza wrażliwość modelu na współliniowość pomiędzy zmiennymi oraz poprawia jego zdolność generalizacji. Model Ridge jest szczególnie odpowiedni dla zbiorów danych takich jak Ames Housing,

gdzie wiele cech opisuje podobne aspekty nieruchomości (np. różne miary powierzchni), a ich wpływ na cenę jest częściowo redundantny.

5.1.3 Regresja ElasticNet

ElasticNet łączy regularyzację L1 (LASSO) oraz L2 (Ridge), co opisuje funkcja celu:

$$\min_{\beta} \left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right).$$

Dzięki temu model jednocześnie:

- redukuje współliniowość (jak Ridge),
- umożliwia częściową selekcję zmiennych (jak LASSO).

ElasticNet jest szczególnie dobrze dopasowany do problemów wysokowymiarowych, gdzie nie wszystkie cechy mają istotny wpływ na zmienną objaśnianą. W kontekście analizowanego zbioru danych pozwala on ograniczyć wpływ mniej informacyjnych predyktorów, zachowując stabilność modelu.

5.1.4 Random Forest

Random Forest jest zespołowym modelem drzew decyzyjnych, w którym predykcja jest średnią wyników wielu drzew trenowanych na losowych podzbiorach danych i cech:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x),$$

gdzie T_b oznacza predykcję b-tego drzewa.

Model ten nie zakłada liniowości zależności i dobrze radzi sobie z interakcjami pomiędzy zmiennymi. Zastosowanie Random Forest umożliwia uchwycenie złożonych relacji pomiędzy cechami nieruchomości, jednak kosztem mniejszej interpretowalności oraz potencjalnie gorszej generalizacji w przypadku ograniczonej liczby obserwacji po czyszczeniu danych.

5.1.5 Gradient Boosting

Gradient Boosting buduje model sekwencyjnie, gdzie każde kolejne drzewo aproksymuje błędy popełnione przez poprzednie predyktory:

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x),$$

gdzie $h_m(x)$ jest nowym drzewem uczonym na resztach, a ν współczynnikiem uczenia.

Model ten charakteryzuje się wysoką elastycznością i zdolnością do modelowania nieliniowych zależności oraz złożonych interakcji pomiędzy zmiennymi. Jest szczególnie użyteczny w problemach, gdzie wpływ cech na zmienną docelową nie ma charakteru addytywnego, co jest typowe dla rynku nieruchomości.

5.1.6 Model zespołowy (Ensemble)

W celu połączenia zalet modeli liniowych i nieliniowych zastosowano model zespołowy zdefiniowany jako średnia predykcji modeli ElasticNet oraz Gradient Boosting:

$$\hat{y}_{ens} = \frac{1}{2}\left(\hat{y}_{EN} + \hat{y}_{GB}\right).$$

Takie podejście pozwala połączyć stabilność i odporność na współliniowość modeli liniowych z elastycznością modeli drzewiastych. Ensemble redukuje ryzyko nadmiernego dopasowania pojedynczego algorytmu oraz zwiększa odporność predykcji na specyficzne cechy danych treningowych.

5.1.7 XGBoost

XGBoost stanowi zaawansowaną implementację gradientowego boostingu, wykorzystującą m.in. regularyzację, ważenie obserwacji oraz optymalizację obliczeń. Model ten został uwzględniony w analizie jako punkt odniesienia dla nowocześniejszych metod boostingowych, jednak ze względu na swoją złożoność oraz większe ryzyko przeuczenia, nie został wybrany jako główny model końcowy.

5.2 Trenowanie modeli

Po zakończeniu etapu eksploracji danych, czyszczenia oraz przygotowania cech, przystąpiono do właściwego procesu trenowania modeli regresyjnych.

Celem tego etapu nie była jeszcze optymalizacja parametrów ani wybór modelu końcowego, lecz rzetelna ocena zachowania różnych algorytmów w identycznych warunkach eksperymentalnych.

5.2.1 Schemat eksperymentu

Wszystkie modele trenowano zgodnie z jednolitym schematem:

- identyczny zestaw cech wejściowych,
- ten sam pipeline przetwarzania danych,
- ta sama transformacja zmiennej docelowej,
- pięciokrotna walidacja krzyżowa.

Takie podejście zapewnia porównywalność modeli i eliminuje wpływ losowego podziału danych na końcowe wnioski.

5.2.2 Walidacja krzyżowa

Do oceny stabilności modeli zastosowano pięciokrotną walidację krzyżową. W każdym przebiegu dane dzielone były na pięć części, z których cztery służyły do trenowania modelu, a jedna do walidacji. Proces ten powtarzano pięciokrotnie, każdorazowo zmieniając zbiór walidacyjny.

Zastosowanie walidacji krzyżowej pozwala:

- ograniczyć ryzyko nadmiernego dopasowania,
- uzyskać bardziej stabilną ocenę jakości,
- lepiej porównać modele o różnej złożoności.

5.2.3 Zakres testowanych modeli

W pierwszym etapie wytrenowano siedem modeli:

- modele liniowe: regresja liniowa, Ridge, ElasticNet,
- modele drzewiaste i zespołowe: Random Forest, Gradient Boosting, XGBoost,
- model hybrydowy oparty na uśrednianiu predykcji.

5.2.4 Wstępne obserwacje

Na etapie trenowania modeli po oczyszczeniu danych zauważalne były istotne różnice między nimi:

- **ElasticNet** osiągał najwyższą jakość predykcji i był najbardziej stabilny w kontekście współliniowości oraz mniej istotnych cech, co czyni go silnym kandydatem do modelu końcowego.
- **Ridge** zachowywał stabilność podobną do ElasticNet, ale nieco gorzej radził sobie z redukcją wpływu mniej informacyjnych zmiennych.
- **Modele drzewiaste** (Random Forest, Gradient Boosting) wykazywały dużą elastyczność i zdolność uchwycenia nieliniowych zależności, jednak były bardziej wrażliwe na obserwacje skrajne i wymagały starannej regularyzacji.
- **Ensemble** łączył mocne strony modeli liniowych i drzewiastych, zapewniając przewidywania o umiarkowanej elastyczności i lepszej odporności na specyficzne outliery.
- **Regresja liniowa** służyła jako model referencyjny – była prosta i interpretowalna, ale ograniczona w uchwyceniu złożonych zależności i wrażliwa na współliniowość.

Szczegółowe wyniki metryk dla wszystkich modeli przedstawiono w kolejnej sekcji.

5.3 Optymalizacja hiperparametrów

Na podstawie wyników uzyskanych w etapie wstępnej oceny jakości modeli wybrano dwa algorytmy do dalszej analizy: **ElasticNet** oraz **Gradient Boosting**. Wybór ten wynikał z ich relatywnie stabilnego zachowania oraz korzystnego kompromisu pomiędzy zdolnością generalizacji a złożonością modelu. Gradient Boosting został uwzględniony jako najlepszy przedstawiciel modeli drzewiastych, wykazując wysoką elastyczność i skuteczność w uchwyceniu nieliniowych zależności w danych.

W niniejszym podrozdziale skoncentrowano się wyłącznie na procesie regularyzacji oraz optymalizacji hiperparametrów. Szczegółowy opis zastosowanego feature engineeringu został przedstawiony we wcześniejszej części pracy.

5.3.1 Cel optymalizacji

Celem optymalizacji hiperparametrów było:

- ograniczenie ryzyka przeuczenia modeli,
- poprawa stabilności predykcji,
- weryfikacja, czy bardziej złożone konfiguracje modeli prowadzą do realnej poprawy jakości.

Proces optymalizacji traktowano również jako eksperyment potwierdzający, czy wcześniejsze wyniki modeli nie były konsekwencją przypadkowego doboru parametrów.

5.3.2 Optymalizacja modelu ElasticNet

W przypadku modelu ElasticNet optymalizacji poddano parametry regularyzacji:

- parametr α, kontrolujący siłę kary regularyzacyjnej,
- parametr *l1_ratio*, określający proporcję pomiędzy regularyzacją L1 i L2.

Do wyszukiwania optymalnej konfiguracji zastosowano metodę przeszukiwania siatki (*GridSearchCV*) z pięciokrotną walidacją krzyżową. Proces ten pozwolił na systematyczne sprawdzenie różnych kombinacji parametrów przy zachowaniu spójnego schematu walidacyjnego.

Uzyskane wyniki wskazały, że silniejszy komponent regularyzacji L1 sprzyja stabilniejszym predykcjom, co potwierdza zasadność stosowania ElasticNet w przypadku danych o wysokiej liczbie cech oraz potencjalnej współliniowości.

5.3.3 Optymalizacja modelu Gradient Boosting

Dla modelu Gradient Boosting przeprowadzono optymalizację kluczowych hiperparametrów wpływających na złożoność i dynamikę uczenia:

- liczby drzew w modelu,
- tempa uczenia (*learning rate*),

- maksymalnej głębokości drzew,
- udziału próbek wykorzystywanych w procesie uczenia.

Podobnie jak w przypadku ElasticNet, zastosowano przeszukiwanie siatki z pięciokrotną walidacją krzyżową. Celem było znalezienie konfiguracji umożliwiającej lepsze uchwycenie złożonych zależności przy jednoczesnym zachowaniu kontroli nad przeuczeniem.

5.3.4 Wyniki optymalizacji

Wyniki optymalizacji hiperparametrów okazały się umiarkowane. W porównaniu z modelami bazowymi oraz wersjami opartymi na pełnych pipeline'ach przetwarzania danych, uzyskana poprawa jakości predykcji była ograniczona, a w części przypadków miała charakter marginalny.

Zaobserwowano, że:

- sama optymalizacja hiperparametrów nie jest w stanie w pełni zrekompensować ograniczeń wynikających z jakości oraz struktury danych wejściowych,
- kluczowy wpływ na jakość predykcji miały wcześniejsze etapy przetwarzania i inżynierii cech,
- zwiększenie złożoności modeli nie zawsze prowadziło do istotnej poprawy zdolności generalizacji.

Uzyskane rezultaty potwierdzają, że skuteczność modeli regresyjnych w dużej mierze zależy od jakości cech wejściowych, natomiast optymalizacja hiperparametrów powinna być traktowana jako etap uzupełniający, a nie substytut, dobrze zaprojektowanego procesu preprocessingu danych.

6 Wyniki i wnioski

6.1 Ocena jakości modeli i podsumowanie wyników

6.1.1 Podsumowanie wyników modeli

Poniżej przedstawiono wyniki siedmiu wytrenowanych modeli regresyjnych po pełnym oczyszczeniu danych. Tabela 2 zawiera metryki MSE, RMSE,

MAE oraz R^2 obliczone na zbiorze walidacyjnym (po cyklicznym czyszczeniu i przygotowaniu danych).

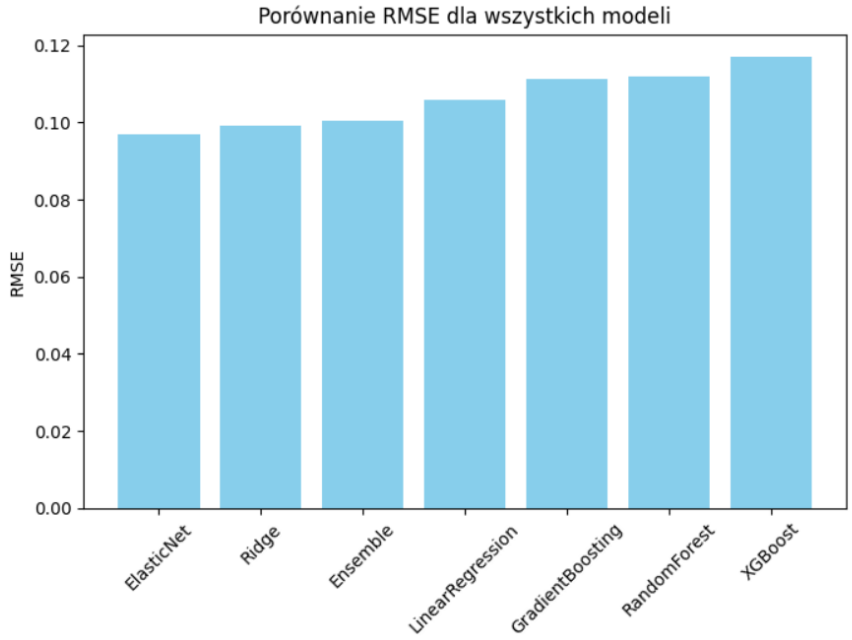
Model	**MSE**	**RMSE**	**MAE**	$\mathbf{R^2}$
ElasticNet	0.009417	0.097043	0.069834	0.892381
Ridge	0.009841	0.099201	0.070536	0.887541
Ensemble	0.010082	0.100410	0.069340	0.884782
LinearRegression	0.011176	0.105716	0.074896	0.872286
GradientBoosting	0.012406	0.111384	0.078077	0.858222
RandomForest	0.012505	0.111824	0.079441	0.857100
XGBoost	0.013703	0.117059	0.077131	0.843406

Tabela 2: Metryki jakości modeli po pełnym przetworzeniu danych.

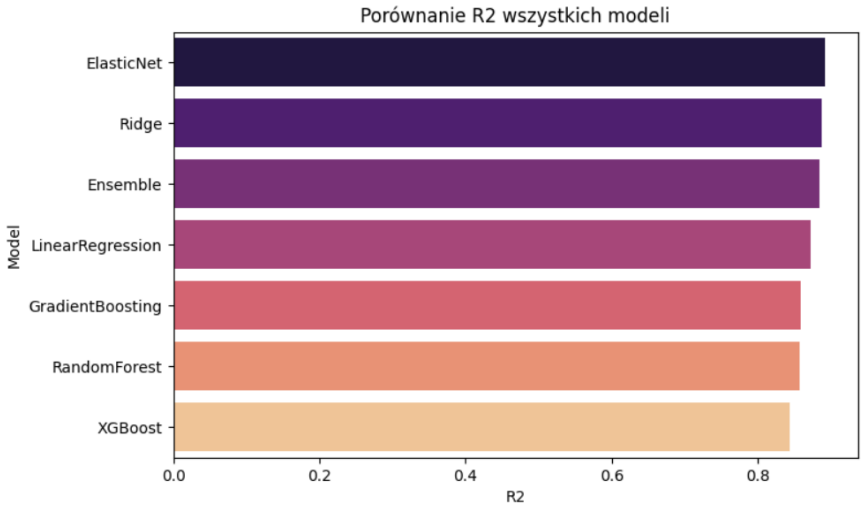
Analiza tabeli wskazuje, że modele liniowe z regularyzacją (ElasticNet, Ridge) osiągają najwyższą wartość R^2 i najniższe RMSE, co sugeruje, że dane mają silny komponent liniowy. Modele drzewiaste wykazują większą wariancję predykcji, lecz nadal radzą sobie z nieliniowymi interakcjami między cechami. Model Ensemble łączy zalety modeli liniowych i nieliniowych, oferując stabilne wyniki predykcyjne.

6.1.2 Wizualizacja jakości modeli

Porównanie RMSE i $\mathbf{R^2}$ Na wykresach słupkowych porównano RMSE oraz R^2 dla wszystkich modeli.



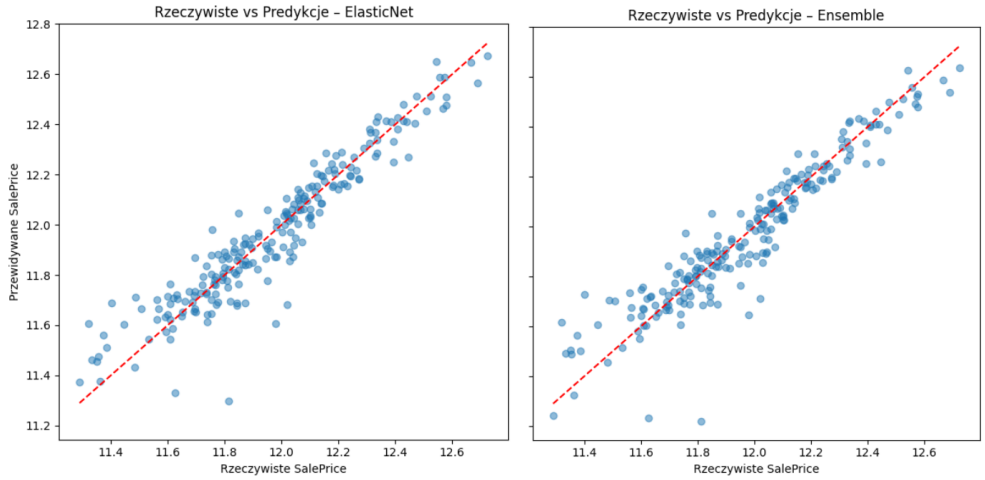
Rys. 11. Porównanie RMSE dla wszystkich modeli. Niższe RMSE oznacza lepsze dopasowanie.



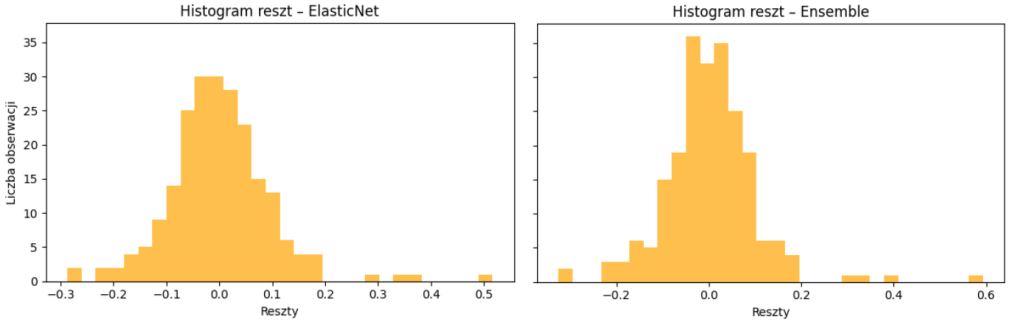
Rys. 12. Porównanie R^2 dla wszystkich modeli. Wyższe R^2 oznacza lepszą zdolność wyjaśniania zmienności danych.

Predykcje vs wartości rzeczywiste Dla modeli ElasticNet oraz Ensemble przygotowano wykresy:

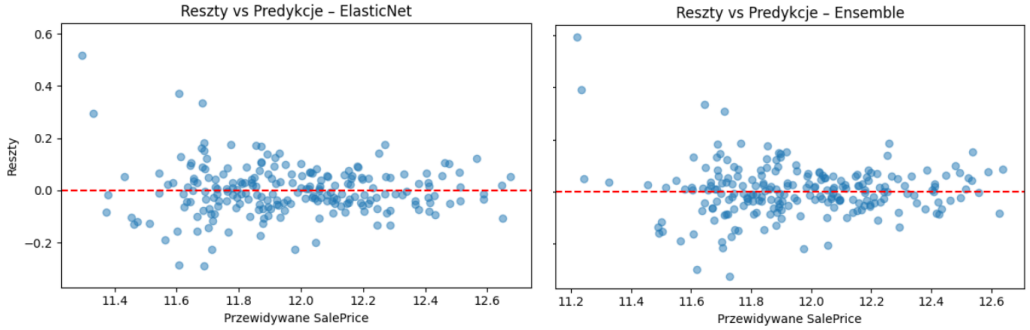
- Scatterplot: wartości rzeczywiste vs predykowane,
- Residual plot: rozkład reszt,
- Reszty vs predykcje: analiza heteroscedastyczności.



Rys. 13. Predykcje vs wartości rzeczywiste dla ElasticNet (lewo) i Ensemble (prawo).



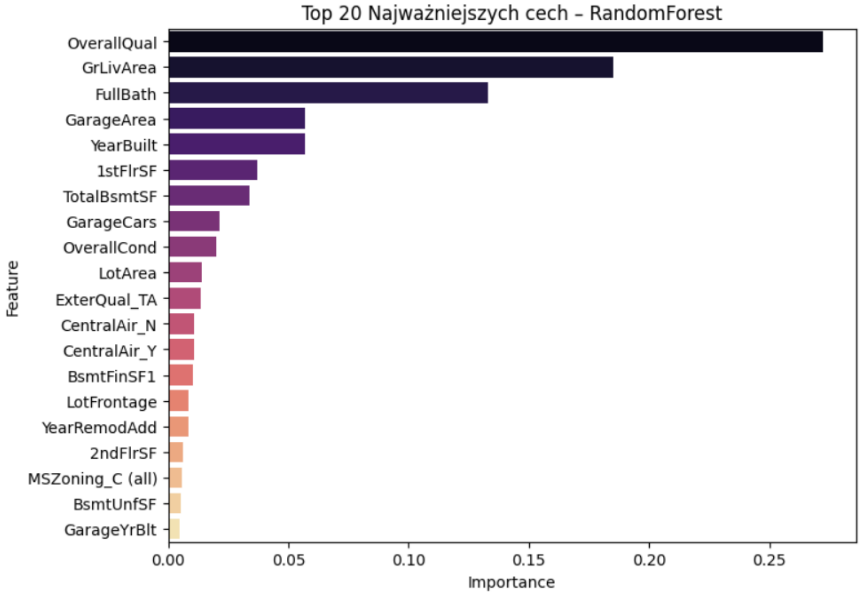
Rys. 14. Analiza reszt dla ElasticNet (lewo) i Ensemble (prawo).



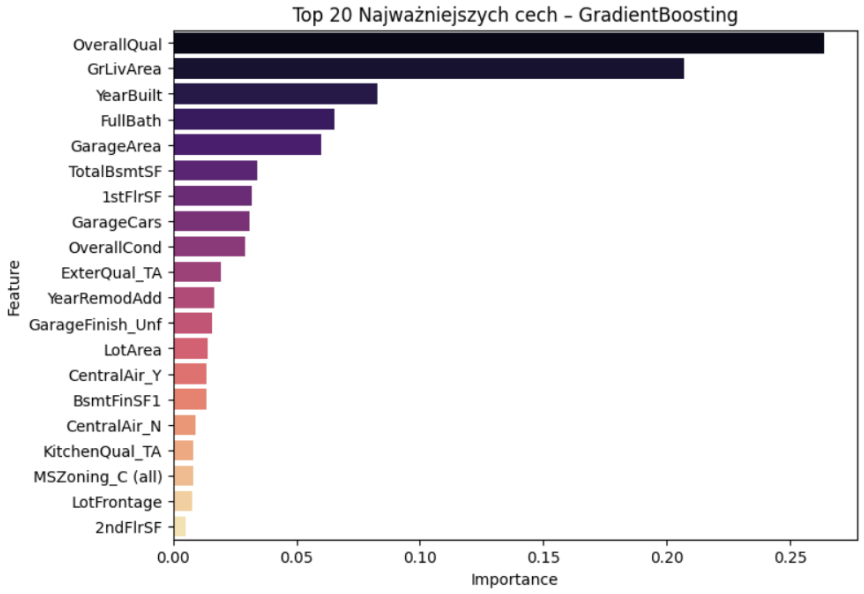
Rys. 15. Reszty vs Predykcje dla ElasticNet (lewo) i Ensemble (prawo).

6.1.3 Najważniejsze cechy w modelach drzewiastych

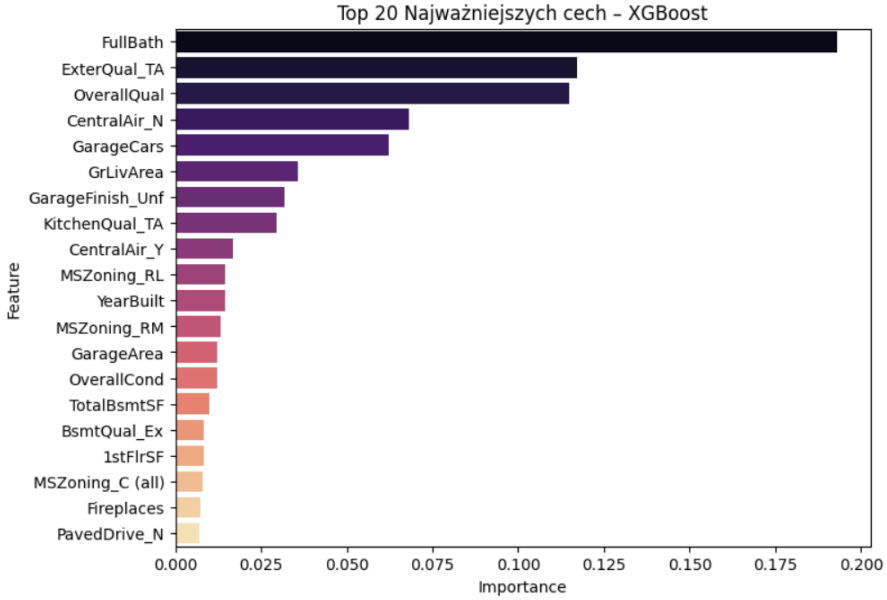
Dla każdego modelu drzewiastego (Random Forest, Gradient Boosting, XGBoost) przygotowano wykres słupkowy przedstawiający Top 20 cech według ich ważności w modelu. Wykresy pozwalają ocenić, które zmienne mają największy wpływ na predykcję ceny sprzedaży domu w poszczególnych algorytmach.



Rys. 16. Top 20 najważniejszych cech w modelu Random Forest.



Rys. 17. Top 20 najważniejszych cech w modelu Gradient Boosting.



Rys. 18. Top 20 najważniejszych cech w modelu XGBoost.

Komentarz do wykresów Wykresy słupkowe dla poszczególnych modeli pozwalają zauważyć, że mimo pewnych różnic w kolejności, kilka cech jest konsekwentnie istotnych we wszystkich modelach. Szczególnie wyróżniają się: *OverallQual*, *GrLivArea* i *FullBath*, co jest zgodne z intuicją dotyczącą czynników wpływających na wartość nieruchomości.

Uśrednione ważności cech – Top 10 Na podstawie wszystkich modeli drzewiastych obliczono średnią ważność cech. Pozwala to wskazać najbardziej kluczowe zmienne wpływające na cenę domu w sposób uogólniony:

Feature	Importance
OverallQual	0.217067
GrLivArea	0.142610
FullBath	0.130554
YearBuilt	0.051349
ExterQual_{TA}	0.050089
GarageArea	0.043149
GarageCars	0.038190
CentralAir_{N}	0.029363
TotalBsmtSF	0.026004
1stFlrSF	0.025783

Tabela 3: Uśrednione ważności cech (Top 10) dla modeli drzewiastych.

Najwyższe wartości wskazują, które zmienne mają największy wpływ na przewidywanie ceny domu w uogólnieniu dla wszystkich trzech modeli. Z analizy wynika, że cechy związane z ogólną jakością domu (OverallQual), powierzchnią użytkową (GrLivArea) oraz liczbą łazienek (FullBath) są najbardziej istotne, co jest zgodne z intuicją i oczekiwaniami dla tego typu danych. Mniej istotne, lecz nadal znaczące cechy, to m.in. rok budowy (YearBuilt), cechy garażu (GarageArea, GarageCars) oraz podstawowe parametry mieszkalne (TotalBsmtSF, 1stFlrSF). Uśrednienie pozwala zredukować wariancję między poszczególnymi modelami i wskazać zmienne kluczowe w sposób stabilny i bardziej ogólny.

Wnioski z analizy cech

- Najważniejsze cechy są w większości spójne między modelami, co potwierdza stabilność predyktorów.

- Modele drzewiaste pozwalają wychwycić zarówno cechy liniowe (np. *OverallQual*, *GrLivArea*), jak i bardziej złożone zależności (np. kombinacje cech infrastruktury i wyposażenia).
- Analiza top 20 cech pozwala zidentyfikować dodatkowe, mniej znaczące predyktory, które mogą mieć lokalny wpływ na cenę nieruchomości.

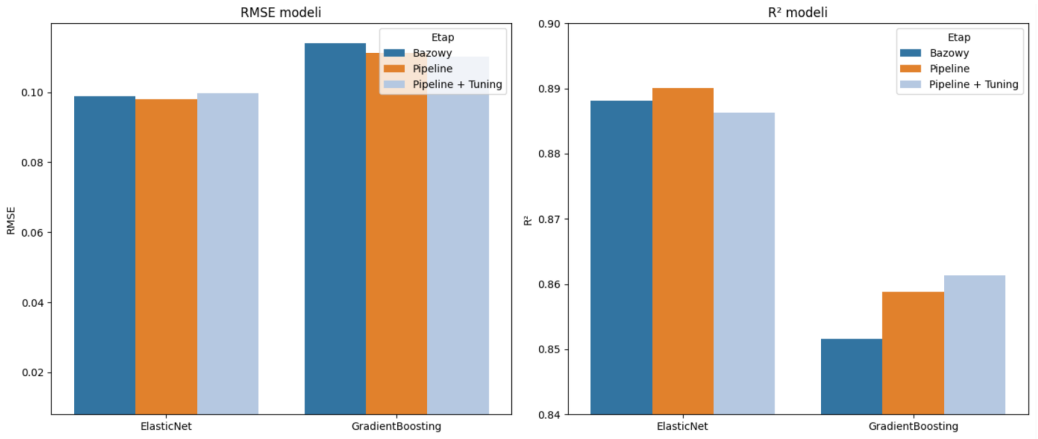
6.1.4 Wyniki po optymalizacji modeli

W celu poprawy jakości predykcji przeprowadzono optymalizację hiperparametrów dla dwóch wybranych modeli: ElasticNet oraz Gradient Boosting. Analizę wykonano na trzech etapach: dla modelu bazowego (B), modelu z pełnym pipeline'em przetwarzania danych (P) oraz modelu po dodatkowej optymalizacji hiperparametrów (P+T).

Tabela 4: Porównanie jakości modeli na kolejnych etapach

Model	**RMSE**			R^2		
	B	P	P+T	B	P	P+T
ElasticNet	0.0989	0.0981	0.0998	0.888	0.890	0.886
Gradient Boosting	0.114	0.111	0.110	0.852	0.859	0.861

B – bazowy, P – pipeline, P+T – pipeline + tuning



Rys. 19. Wykres słupkowy wartości RMSE dla modeli ElasticNet i Gradient Boosting na kolejnych etapach

Na podstawie wyników przedstawionych w tabeli 4 można sformułować następujące obserwacje.

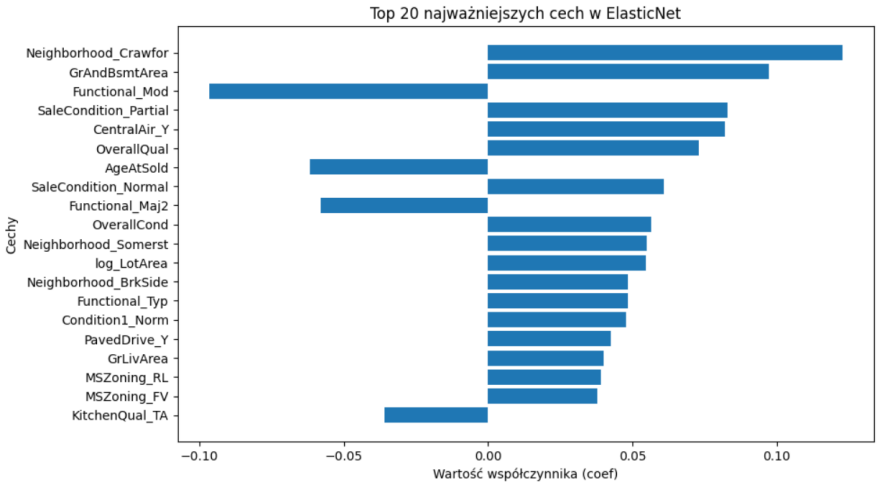
W przypadku modelu **ElasticNet** zastosowanie pipeline'u przetwarzania danych przyniosło niewielką, lecz spójną poprawę jakości predykcji względem wersji bazowej. Spadek wartości RMSE oraz wzrost współczynnika determinacji R^2 wskazują, że transformacje logarytmiczne, inżynieria cech oraz standaryzacja miały pozytywny wpływ na stabilność modelu. Dodatkowa optymalizacja hiperparametrów nie doprowadziła jednak do dalszej poprawy jakości — wzrost RMSE i obniżenie wartości R^2 sugerują, że zwiększona regularyzacja (l1_ratio = 0.9) była zbyt agresywna i usunęła istotne cechy, pogorszając zdolność generalizacji modelu.

Dla modelu **Gradient Boosting** wpływ zastosowania pipeline'u był wyraźniejszy niż w przypadku ElasticNet. Wersja pipeline'owa osiągnęła znaczną poprawę zarówno pod względem RMSE, jak i R^2 w porównaniu do modelu bazowego. Potwierdza to wysoką wrażliwość modeli drzewiastych na jakość przetwarzania danych wejściowych oraz dodatkowe cechy opisujące wiek nieruchomości i sezonowość sprzedaży. W przeciwieństwie do ElasticNet, dalszy tuning hiperparametrów poprzez GridSearchCV przyniósł dodatkową, choć niewielką, poprawę zarówno RMSE, jak i R^2, co wskazuje na skuteczność optymalizacji parametrów takich jak `n_estimators`, `learning_rate` i `max_depth` dla tego typu modelu.

Uzyskane wyniki prowadzą do istotnego wniosku, że **największy wpływ na poprawę jakości predykcji miało zastosowanie odpowiedniego pipeline'u z inżynierią cech** (transformacje logarytmiczne, cechy czasowe, agregaty), który poprawił wyniki obu modeli. Optymalizacja hiperparametrów okazała się skuteczna dla Gradient Boosting, ale nieefektywna dla ElasticNet, gdzie zbyt silna regularyzacja pogorszyła wyniki. W konsekwencji, jako najlepsze modele wybrano: **ElasticNet w wersji pipeline'owej bez tuningu** (RMSE = 0.098, $R^2 = 0.890$) oraz **Gradient Boosting w wersji pipeline + tuning** (RMSE = 0.110, $R^2 = 0.861$).

6.1.5 Najważniejsze cechy w modelu ElasticNet po zastosowaniu pipeline

Dla modelu liniowego ElasticNet z pełnym pipeline przygotowano wykres słupkowy przedstawiający 20 cech o największym wpływie na predykcję ceny sprzedaży domu. Wartość współczynnika (coef) wskazuje kierunek i siłę wpływu danej cechy na przewidywaną wartość.



Rys. 20. Top 20 najważniejszych cech w modelu ElasticNet z zastosowanym pipeline.

Komentarz do wyników Analiza wykresu pokazuje, że najważniejsze cechy w modelu ElasticNet z pipeline różnią się częściowo od modeli drzewiastych (Random Forest, Gradient Boosting, XGBoost). W modelach drzewiastych dominują cechy związane z ogólną jakością domu, powierzchnią mieszkalną, liczbą łazienek czy wielkością garażu. Natomiast ElasticNet lepiej uwzględnia cechy skategoryzowane oraz transformowane, takie jak lokalizacja (`Neighborhood_Crawfor`), warunki sprzedaży (`SaleCondition_Partial`) czy różne wskaźniki funkcjonalności (`Functional_Mod`, `Functional_Maj2`).

Wartości współczynników wskazują również kierunek wpływu: cechy o wartości ujemnej obniżają przewidywaną cenę, natomiast dodatnie – podnoszą ją. W porównaniu z modelami drzewiastymi, ElasticNet z pipeline daje bardziej wyważoną ocenę cech i lepiej uwzględnia subtelne efekty zmiennych kategorycznych, podczas gdy modele drzewiaste koncentrują się na najbardziej oczywistych i bezpośrednich determinantach ceny.

6.2 Finalne predykcje na zbiorze testowym

Po wyborze najlepszych modeli przeprowadzono finalne predykcje na zbiorze testowym. Dla obu modeli zastosowano odwrotną transformację logarytmiczną (`np.expm1()`), aby uzyskać rzeczywiste ceny w dolarach.

Tabela 5: Przykładowe predykcje cen nieruchomości dla pierwszych 5 obserwacji ze zbioru testowego

Id	Gradient Boosting	ElasticNet
1461	\$122,931	\$116,069
1462	\$160,576	\$163,666
1463	\$181,861	\$182,368
1464	\$190,381	\$202,431
1465	\$193,347	\$182,299

Tabela 6: Statystyki opisowe finalnych predykcji na zbiorze testowym

Statystyka	Gradient Boosting	ElasticNet
Średnia	\$173,562	\$179,793
Mediana	\$155,938	\$159,647

Model Gradient Boosting przewiduje średnio niższe ceny nieruchomości (średnia różnica około \$6,200), co może wynikać z jego większej wrażliwości na ekstremalne wartości oraz bardziej konserwatywnego podejścia wynikającego z zastosowanej regularyzacji (`subsample` = 0.8). ElasticNet, mimo wyższej średniej, wykazuje podobny rozkład predykcji, co potwierdza spójność obu modeli.

6.3 Wnioski z analizy

Celem projektu *house-prices-advanced-regression-analysis* było przeprowadzenie kompletnej analizy regresyjnej służącej do predykcji cen sprzedaży nieruchomości, obejmującej pełny proces analityczny: eksplorację danych, czyszczenie i przygotowanie zbioru, inżynierię cech, budowę pipeline'ów przetwarzania danych, trenowanie modeli oraz ich porównanie pod względem jakości predykcji.

Analiza eksploracyjna danych wykazała, że zmienna objaśniana cechuje się silną prawoskośnością, a w zbiorze występują obserwacje odstające związane głównie z bardzo dużą powierzchnią użytkową. Zastosowanie transformacji logarytmicznej zmiennej docelowej oraz selektywne usunięcie obserwacji skrajnych znacząco poprawiło stabilność modeli i jakość dopasowania, co potwierdziło zasadność etapu wstępnego czyszczenia danych.

Porównanie wielu algorytmów regresyjnych wykazało, że najlepsze wyniki osiągnęły dwa podejścia: model liniowy **ElasticNet** z inżynierią cech (RMSE = 0.098, R^2 = 0.890) oraz model drzewiasty **Gradient Boosting** z optymalizacją hiperparametrów (RMSE = 0.110, R^2 = 0.861). ElasticNet osiągnął nieznacznie lepsze wyniki dzięki transformacjom logarytmicznym i nowo utworzonym cechom czasowym, potwierdzając, że po odpowiednim przygotowaniu danych relacje pomiędzy zmiennymi a ceną mają w dużej mierze charakter liniowy. Zastosowanie regularyzacji pozwoliło skutecznie ograniczyć wpływ współliniowości.

Kluczowym elementem sukcesu obu modeli okazała się **inżynieria cech**. Utworzono szereg nowych zmiennych:

- transformacje logarytmiczne cech o skośnym rozkładzie (`LotArea`, `LotFrontage`, `MasVnrArea`, `WoodDeckSF`),
- cechy czasowe opisujące wiek nieruchomości (`AgeAtSold`, `YearsSinceRemod`, `HouseAge`, `GarageAge`),
- kodowanie cykliczne miesiąca sprzedaży (sin/cos) oraz zmienna binarna dla sezonu wysokiej sprzedaży,
- cechy agregowane: łączna powierzchnia mieszkalna, całkowita liczba łazienek, wynik jakości.

Modele drzewiaste, takie jak Gradient Boosting, wykazały większą elastyczność w modelowaniu złożonych zależności i interakcji pomiędzy cechami. Gradient Boosting szczególnie skorzystał z optymalizacji hiperparametrów (zwiększenie `n_estimators` do 500, `subsample` = 0.8), co przyniosło dalszą poprawę wyników. W przypadku ElasticNet optymalizacja hiperparametrów okazała się jednak nieefektywna – zwiększona regularyzacja (l1_ratio = 0.9) pogorszyła wyniki poprzez usunięcie istotnych cech.

Zastosowanie pipeline'ów przetwarzania danych miało istotny wpływ na poprawę jakości predykcji we wszystkich analizowanych modelach. Standaryzacja cech numerycznych, kodowanie zmiennych kategorycznych (OneHotEncoder) oraz spójny proces przygotowania danych okazały się kluczowe dla uzyskania stabilnych i porównywalnych wyników. Największą poprawę przyniosła jednak inżynieria cech – w przypadku ElasticNet RMSE spadł z 0.099 (baseline) do 0.098 (pipeline), a w przypadku Gradient Boosting z 0.114 do 0.111.

Podsumowując, przeprowadzona analiza potwierdza, że **największy wpływ na skuteczność modeli miały transformacje logarytmiczne oraz inżynieria cech czasowych i agregowanych**, a nie sama złożoność algorytmu czy optymalizacja hiperparametrów. Oba finalne modele – ElasticNet (pipeline bez tuningu) oraz Gradient Boosting (pipeline z tuningiem) – osiągnęły porównywalne, wysokie wyniki i mogą być wykorzystane zamiennie w zależności od preferencji dot. interpretowalności (ElasticNet) lub zdolności do modelowania nieliniowości (Gradient Boosting).

7 Struktura projektu

Projekt **house-prices-advanced-regression-analysis** został zorganizowany w następujący sposób:

- **.idea**/ – pliki konfiguracyjne środowiska IDE (PyCharm/IntelliJ).
- **.ipynb_checkpoints**/ – automatycznie tworzone checkpointy notebooków Jupyter.
- **data**/ – główny folder z danymi, podzielony na:
  - **raw**/ – surowe dane pobrane z Kaggle,
  - **processed**/ – dane po wstępnej obróbce i oczyszczeniu (`train_postprocessed.csv`, `test_postprocessed.csv`),
  - **predictions**/ – zapisy wyników predykcji modeli (`model_predictions.csv`),
  - **data_description.txt** – szczegółowy opis zmiennych i źródeł danych,
  - **data.py** – skrypt do czyszczenia i przygotowania danych.
- **env**/ – środowisko wirtualne projektu (nieśledzony w repozytorium).
- **notebooks**/ – notebooki Jupyter zawierające cały proces analizy:
  - **00_eda.ipynb, 01_eda.ipynb** – eksploracyjna analiza danych (EDA), wizualizacje rozkładów i korelacji,
  - **02_models.ipynb** – trenowanie modeli bazowych (LinearRegression, Ridge, ElasticNet, RandomForest, GradientBoosting, XGBoost),

  - **03_resultsAnalysis.ipynb** – analiza wyników, porównanie modeli i wizualizacje predykcji,
  - **04_feature_engineering_optimization.ipynb** – zaawansowana inżynieria cech, selekcja zmiennych oraz optymalizacja hiperparametrów,
  - **best_enet_pipeline.pkl, best_gb_pipeline.pkl** – zapisane najlepsze modele (pipeline'y) gotowe do predykcji.

- **plots**/ – folder zawierający wykresy i wizualizacje generowane w trakcie analizy.

- **reports**/ – dokumentacja projektu w formacie LaTeX, zawierająca:
  - szczegółowy opis metodologii,
  - analizę statystyczną,
  - porównanie modeli,
  - techniki inżynierii cech,
  - wnioski i rekomendacje.

- **src**/ – moduły źródłowe zawierające funkcje i klasy wykorzystywane w projekcie.

- **text-data**/ – dodatkowe pliki tekstowe z opisami danych.