Statistical learning w praktyce

9 lipca 2025

1 Wybór danych i postawienie problemu

Celem naszej analizy jest stworzenie modelu regresji przewidującego **ceny domów** na podstawie dostępnych cech nieruchomości, takich jak: powierzchnia, lokalizacja, liczba sypialni. Oprócz samej predykcji, istotna dla nas jest **analiza danych**, gdyż chcemy dokładnie poznać strukturę naszego problemu.
Dane mają ponad **10 lat**, więc predykcja może nie być w pełni użyteczna. Jednak gdy poznamy **wewnętrzną strukturę danych**, dowiemy się, jakie zmienne były istotne i jaki miały wpływ na możliwości predykcyjne. Zdobyta wiedza może zostać wykorzystana w analizie całkowicie nowych danych. Z tych powodów decydujemy się **nie wydzielać zbioru testowego i treningowego**, tylko porównywać błąd **RMSE modeli** między sobą za pomocą **walidacji krzyżowej** na całym zbiorze danych.
Rozważamy **zbiór danych** dotyczących domów, które zostały sprzedane od **maja 2014 roku do maja 2015 roku** w **Hrabstwie King** w stanie Waszyngton w USA. Dane te zostały zebrane przez urząd tego hrabstwa. Więcej informacji jest dostępnych pod tym linkiem: 2014-15 Home Sales in King County. Zbiór ten został **zmodyfikowany pod nasze potrzeby**, między innymi zmieniono jednostki z amerykańskich na europejskie, np. *square feet* na *square meters*, aby łatwiej można było przeprowadzić analizę.
Posiadamy łącznie **13 603 obserwacje**. Dane są dostępne pod tym adresem: House Price Prediction Treated Dataset.

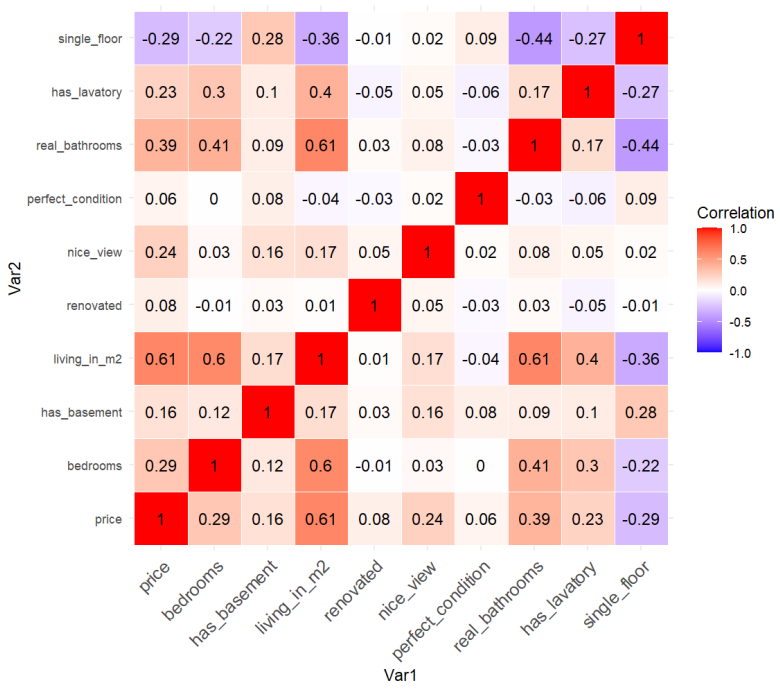
Rysunek 1: Dokładna lokalizacja każdej obserwacji

2 Eksploracja i przygotowanie danych

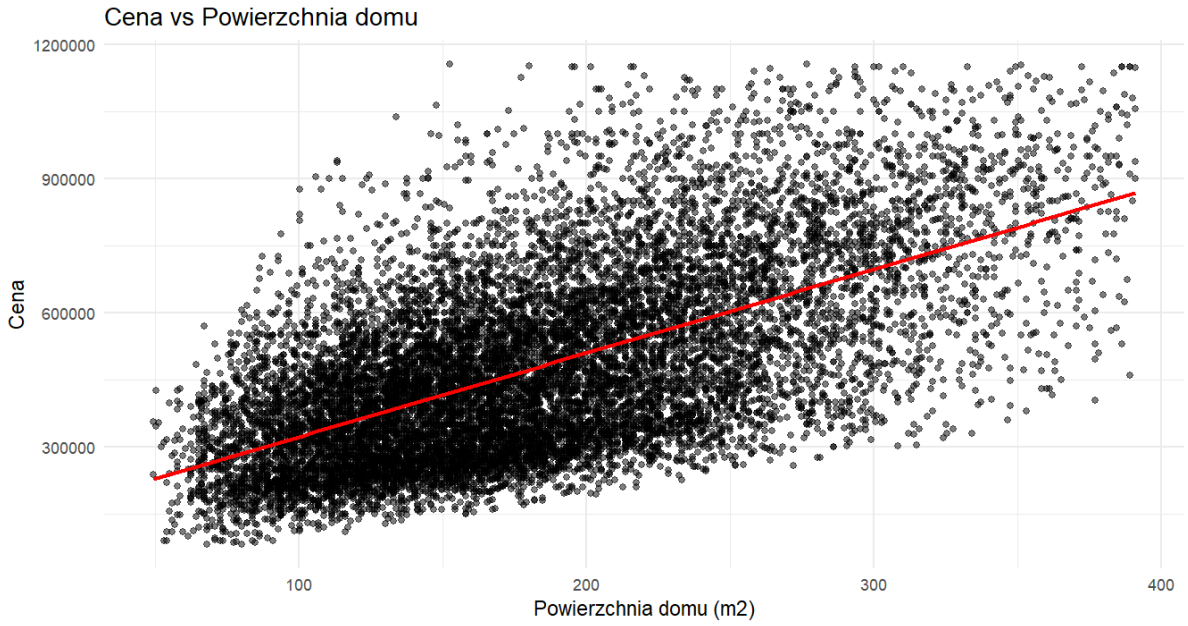
W naszym zbiorze znajduje się **14** zmiennych. Poniżej mamy opis każdej z nich.

- date: data sprzedaży domu,
- price: cena, za którą sprzedano dom,
- bedrooms: liczba sypialni,
- grade: ocena jakości pod względem konstrukcji i zaprojektowania od **1** do **5**,
- has_basement: czy dom posiada piwnicę `(TRUE, FALSE)`,
- living_in_m2: metraż domu w metrach kwadratowych,
- renovated: czy dom było remontowany `(TRUE, FALSE)`,
- nice_view: ładny widok z domu `(TRUE, FALSE)`,
- perfect_condition: idealny stan domu `(TRUE, FALSE)`,
- real_bathrooms: liczba łazienek,
- has_lavatory: czy jest osobna toaleta oprócz łazienki `(TRUE, FALSE)`,
- single_floor: czy dom jest jednopiętrowy `(TRUE, FALSE)`,
- month: miesiąc, w którym dom został sprzedany,
- quartile_zone: rozkład kodów pocztowych od **1** do **4** gdzie **1** oznacza tańszą okolice domów a **4** droższą.

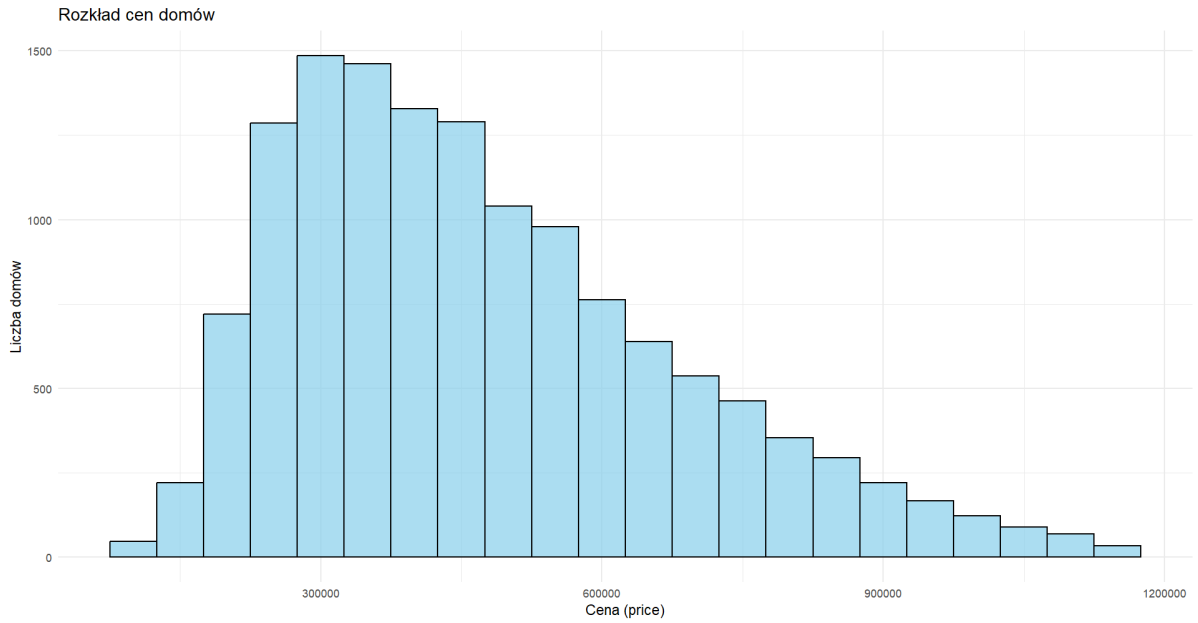
Przeglądamy nasze zmienne i wykonujemy niezbędne czynności potrzebne do analizy danych. W naszym projekcie zmienną objaśnianą jest oczywiście zmienna `price`. Zmienne `grade` oraz `quartile_zone` kodujemy jako zmienne jakościowe. Zmienne zawierające wartości `TRUE, FALSE` zmieniamy na wartości `1,0`. Ponadto usuwamy całe kolumny `date` i `month` ponieważ są one mało interesujące z naszego punktu widzenia. Na koniec upewniamy się, że nasze dane nie zawierają wartości pustych.
Po przygotowaniu danych możemy przejść do eksploracji. Z wyrysowanej mapy ciepła z wartościami korelacji widzimy, że zmienną najbardziej skorelowaną ze zmienną `price` jest `living_in_m2`. Należy pamiętać, że korelacja jest policzona między zmiennymi numerycznymi więc nie możemy zapomnieć o naszych zmiennych jakościowych, które na *Rysunku 2* nie są uwzględnione.
Dalej na *Rysunku 3* zauważamy, że pomiędzy ceną a powierzchnią może istnieć zależność liniowa. Ponadto przyglądamy się rozkładom cen (*Rysunek 4*), powierzchni (*Rysunek 5*) czy cen z uwzględnieniem podziału na strefy (*Rysunek 6*).



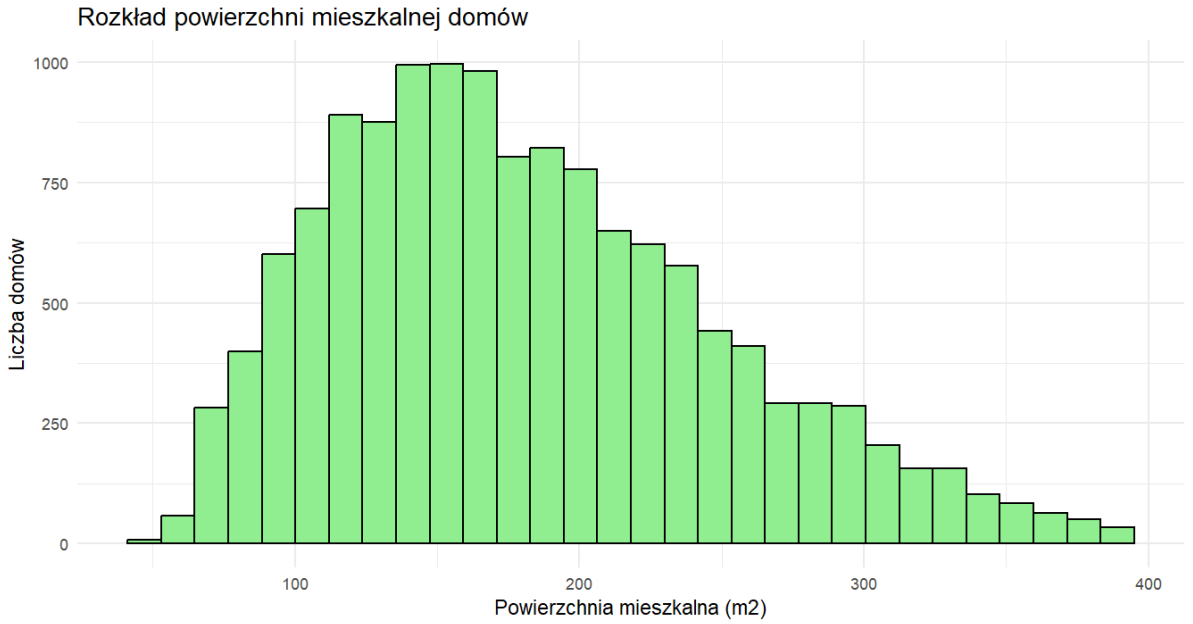
Rysunek 2: Mapa ciepła z wartościami korelacji



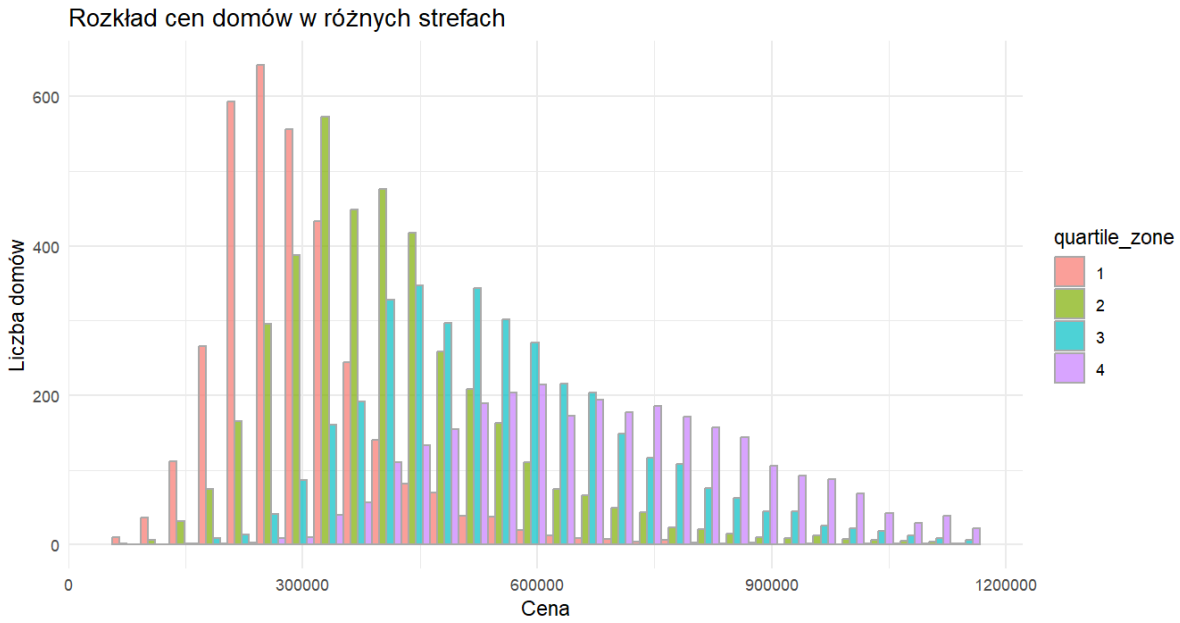
Rysunek 3: Cena domu i powierzchnia



Rysunek 4: Rozkład cen domów



Rysunek 5: Rozkład powierzchni mieszkalnej domów



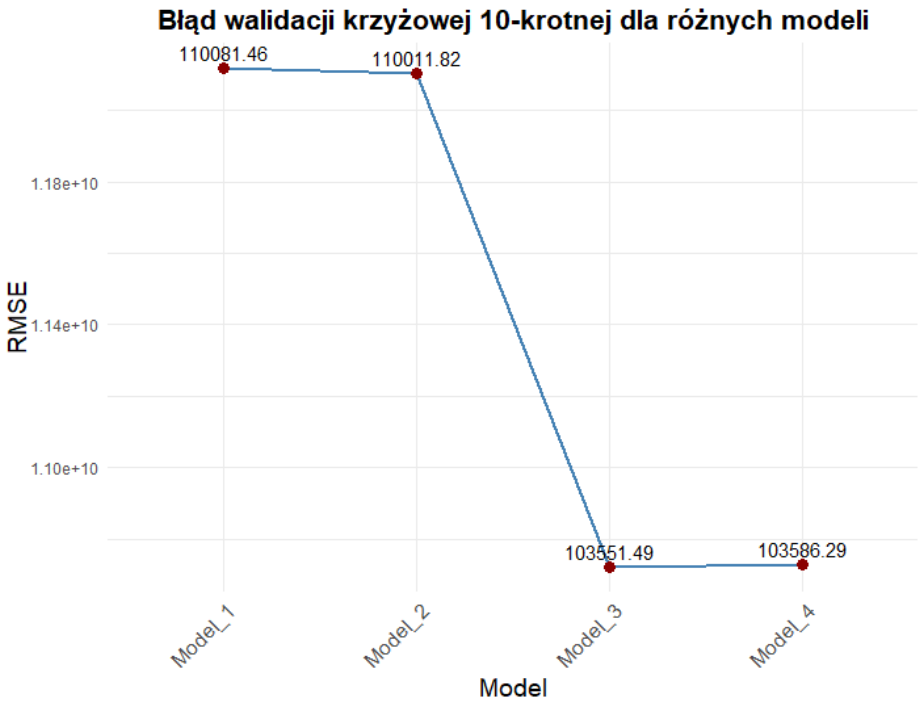
Rysunek 6: Rozkład cen domów w różnych strefach

3 Budowanie modeli

W tej sekcji zajmiemy się budowaniem modeli. Wykorzystamy do tego poznane na zajęciach metody.

3.1 Model regresji liniowej

Stworzymy kilka modeli liniowych na początku wybierając zmienne objaśniające według uznania. **Model_1** zawiera zmienne `grade`, `quartile_zone`, `living_in_m2`. **Model_2** zawiera te zmienne co **Model_1** oraz dodatkowo zmienne z wyższą korelacją czyli `bedrooms`, `real_bathrooms`, `single_floor`. **Model_3** będzie modelem pełnym a **Model_4** pełnym bez zmiennej `bedrooms`, bo z podsumowania wynika, że ta zmienna może być nieistotna.
Następnie robimy **10-krotną** walidację krzyżową na całym zbiorze i wyświetlamy błędy RMSE dla każdego modelu (*Rysunek 7*).



Rysunek 7: Błędy walidacji krzyżowej

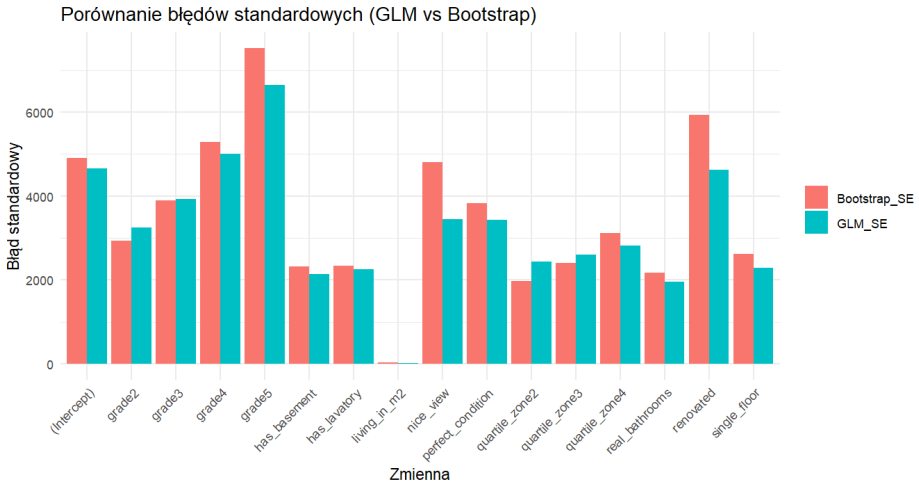
Jak możemy zauważyć to najniższe błędy są dla **Model_3** oraz **Model_4** co może sugerować, że raczej będziemy celować w modele pełne lub prawie pełne.

3.2 Bootstrap

Teraz wykorzystamy bootstrap do estymacji odchylenia standardowego współczynników modelu regresji liniowej. Decydujemy się na **Model_4** bo ma najniższy błąd. Porównując wartości współczynników z funkcji `glm` oraz z bootstrapu widzimy, że nie różnią się znacznie od siebie. Tak samo odchylenia standardowe są praktycznie takie same (*Rysunek 8*) i są małe w odniesieniu do wartości współczynników. Na tej podstawie możemy stwierdzić, że nasz model jest stabilny.

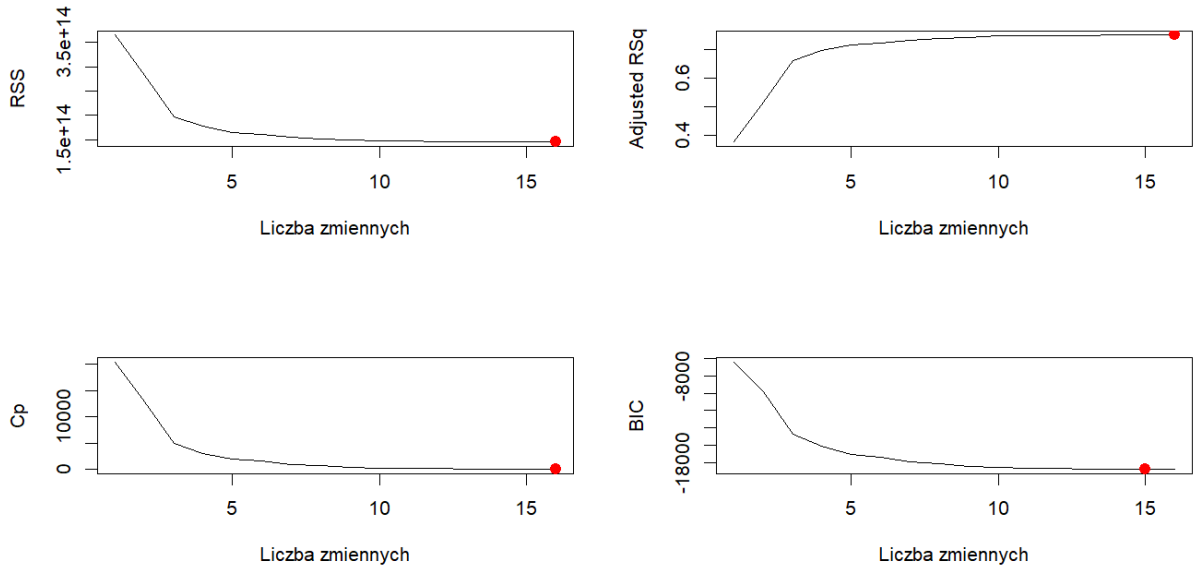
3.3 Wybór zmiennych niezależnych

Na tym etapie chcemy postawić sobie pytanie czy na pewno **Model_4** jest najlepszy? Przecież wybieraliśmy tamte modele losowo. Dlatego w tym podrozdziale skupimy się na tym ile zmiennych powinno trafić do naszego finalnego modelu i które zmienne to powinny być. Na poniższym rysunku (*Rysunek 9*) możemy zobaczyć jaka liczba zmiennych jest optymalna dla miar takich jak:
`RSS, Adjusted R-squared, Mallows's Cp, BIC.` Wynika z tego, że najlepsze modele mają **15** lub **16** zmiennych. Następnie przeprowadzamy selekcję krokową (`forward, backward`). Końcowy wniosek jest



Rysunek 8: Porównanie odchyleń standardowych glm i bootstrapu

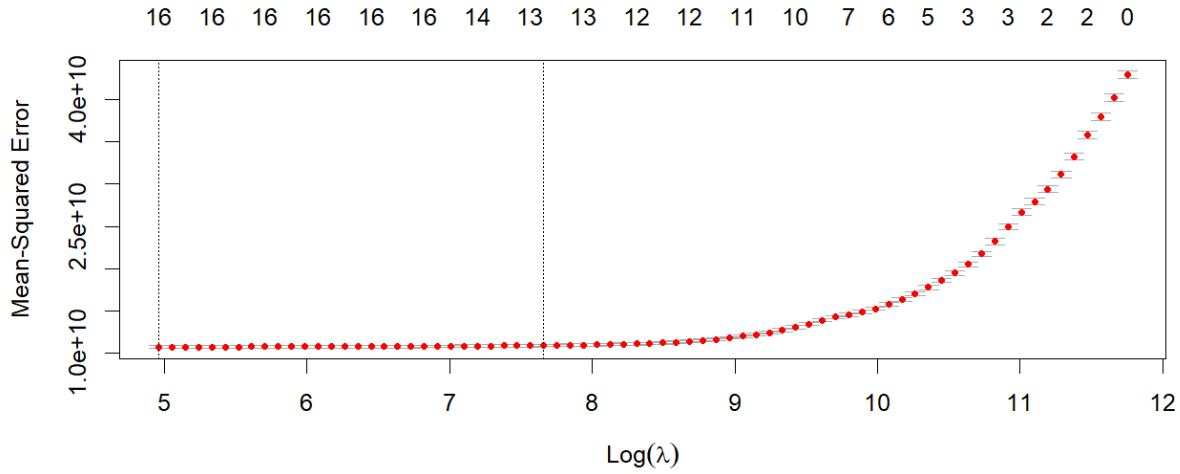
taki, że każda z metod `full, fwd, bdw` dla **15** zmiennych wyrzuciła dokładnie tę samą zmienną `bedrooms` a model z **16** zmiennymi po prostu jest pełny.
Alternatywnie możemy spróbować znaleźć najlepszy model metodą walidacji krzyżowej, tzn. dla każdej liczby zmiennych policzyć błąd walidacyjny z 10-krotnej walidacji krzyżowej i wybrać model o najniższym błędzie. Najniższy błąd wychodzi dla modelu o 16 zmiennych (model pełny) i RMSE wynosi dokładnie 103 569 (**Model_5**).



Rysunek 9: Liczba zmiennych względem odpowiedniej miary

3.4 Regresja grzbietowa i regresja lasso

W tym podrozdziale rozważymy metody ridge i lasso regression czyli modyfikacje metody najmniejszych kwadratów, które mają za zadanie lepiej dobrać współczynniki w modelu regresji liniowej poprzez wprowadzenie kary za zbyt duże wartości β w rozważanym modelu. Najpierw dla regresji grzbietowej za pomocą walidacji krzyżowej szukamy optymalnego parametru λ, siatka dla parametru λ jest stworzona automatycznie przez funkcję `cv.glmnet`. Dla optymalnej lambdy błąd RMSE wynosi 104 762 (**Model_6**). Dla regresji lasso robimy analogicznie i dla optymalnej lambdy błąd RMSE wynosi 104 602 (**Model_7**). Warto zaznaczyć (*Rysunek 10*), że błąd dla **13** zmiennych może być niewiele mniejszy a model łatwiejszy w interpretacji.



Rysunek 10: Regresja lasso

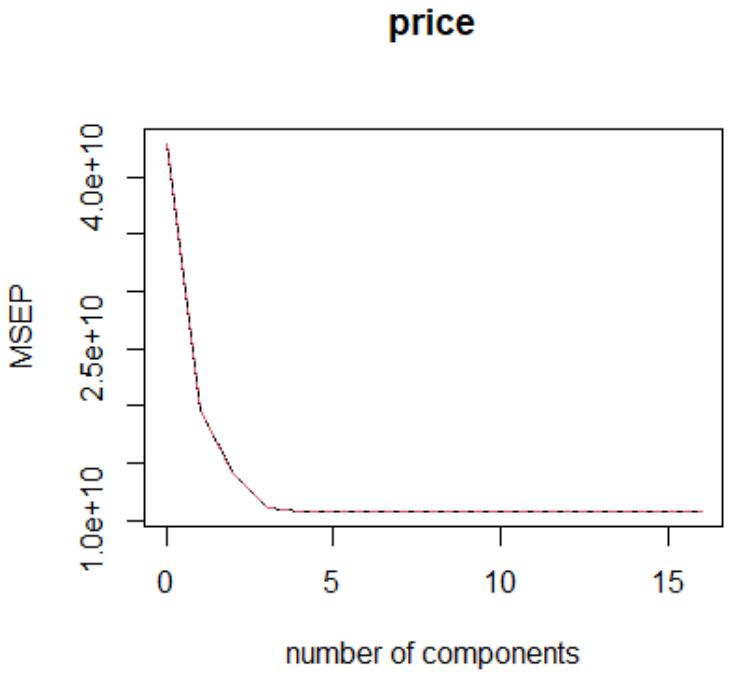
3.5 Redukcja wymiarów

Metoda redukcji wymiarów będzie mało przydatna w naszym przypadku ponieważ dostajemy całkiem nowe zmienne przez co tracimy interpretowalność ale sprawdzimy czy błąd będzie mniejszy niż w pozostałych metodach.

Najpierw rozważmy metodę regresji składowych głównych (PCR). Używamy funkcji `pcr`, która za pomocą walidacji krzyżowej pomoże dam dobrać optymalną liczbę składowych głównch. Najniższe błędy dostajemy gdy liczba zmiennych jest równa **14**, **15** lub **16** i dla tej liczby zmiennych można zauważyć też przeskok w wyjaśnialności wariancji z 50% na około 75%. Błąd RMSE policzony na całym zbiorze dla **14** zmiennych wynosi 104 785 (**Model_8**). Nie zyskaliśmy praktycznie nic na błędzie i dodatkowo nie udało się znacznie zmniejszyć liczby zmiennych (16 $\rightarrow$ 14).

Kolejna metoda redukcji wymiarów to metoda cząstkowych najmniejszych kwadratów (PLS). Tutaj błąd uzyskany za pomocą walidacji krzyżowej jest najmniejszy dla 13 zmiennych ale warto zauważyć, że błąd dla **4** zmiennych będzie tylko niewiele większy (*Rysunek 11*). Dlatego decydujemy się wybrać model z

liczbą zmiennych równą **4** i błąd RMSE wynosi wtedy 103 832 (**Model_9**) przy 75% wyjaśnialności wariancji zmiennej `price`.



Rysunek 11: Liczba zmiennych w metodzie PLS

3.6 Drzewa regresyjne

Przechodzimy teraz do metod drzewiastych, których zaletą jest dobra wizualizacja i co za tym idzie prosta interpretacja. Tworzymy duże drzewo pozwalając mu się nawet przeuczyć ale modele zachowują się w taki sposób, że optymalna liczba liści jest zawsze dla największego drzewa. Nie ma etapu przycinania bo przycinanie nic nie polepsza. Przykładowo dla parametru `mindev` = 0.001 duże drzewo ma **37** liści i dla 37 liści jest najmniejsza dewiancja. Oznaczamy to jako **Model_10** i jego błąd RMSE wynosi 107 041. Przy jeszcze większym drzewie zawierającym 1255 liści błąd wynosi około 85 000 ale zdajemy sobie sprawę, że jest to drzewo przeuczone. Podejrzewamy, że jest to kwestia skomplikowanych danych i przez to mniejsze drzewa powodują dużą utratę informacji i zwiększenie błędu. Zdecydowaliśmy żeby nie pokazywać rysunków takich drzew gdyż są one nieczytelne.

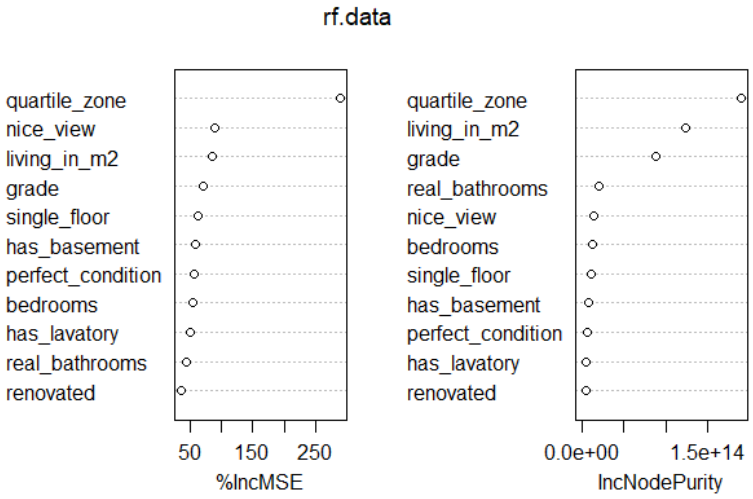
3.7 Bagging

Teraz wykorzystamy metodę bagging'u, która wykorzystując ideę boostrapu pomoże nam zmniejszyć wariancję modelu. Zatem ustawiamy zmienną `mtry` = 11 bo tyle jest zmiennych w naszym modelu (nie

wlicza się tutaj podział spowodowany zmiennymi typu factor). Otrzymujemy **Model_11** z błędem RMSE równym 107 513. Błąd jest porównywalny z drzewem regresyjnym ale większy od poprzednich metod.

3.8 Random Forest

Sprawdzamy jak z naszym problemem poradzi sobie las losowy, który jest uogólnieniem bagging'u. Wykorzystujemy tę samą funkcję co dla bagging'u tylko z parametrem `mtry` = 3. Dostajemy **Model_12** z błędem 100 698. Z wykresu `varImpPlot` widzimy, że zmiennymi najbardziej istotnymi są kolejno `quartile_zone`, `living_in_m2`, `nice_view`, `grade`.



Rysunek 12: Wykres istotności zmiennych

Lewa część *Rysunku 12* przedstawia, jak bardzo średni błąd (MSE) modelu wzrasta, gdy wartości danej zmiennej zostaną zamienione losowo (permute). Zmienne o wyższym IncMSE są bardziej istotne dla predykcji modelu. Prawa część *Rysunku 12* przedstawia miarę poprawy czystości węzłów wynikającą z podziałów drzewa przy użyciu danej zmiennej.

3.9 Boosting

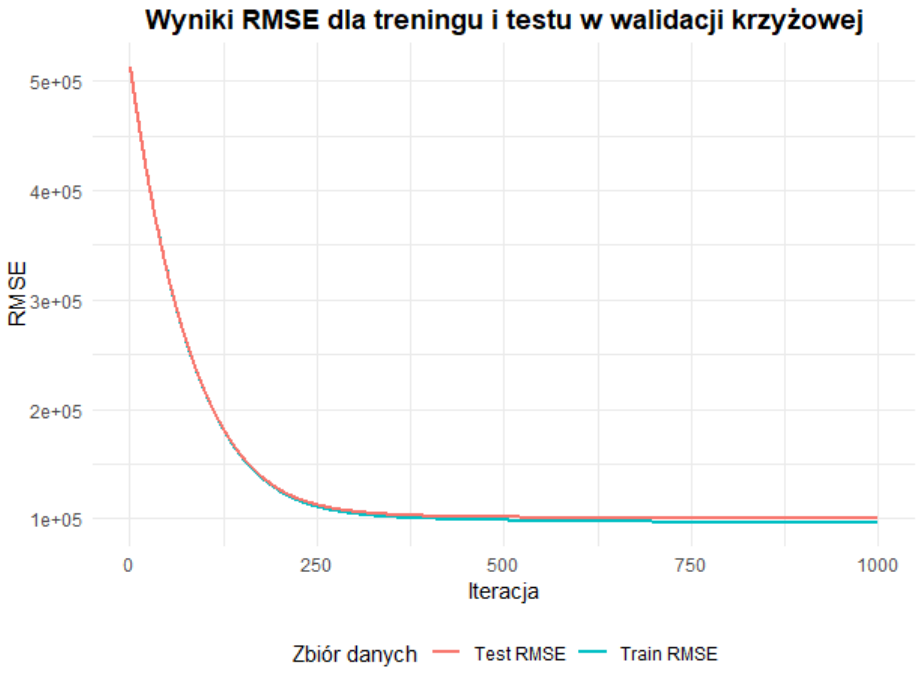
Wykorzystamy metodę boosting'u, która polega na sekwencyjnym budowaniu wielu drzew w sposób zależny od siebie. To znaczy każde kolejne drzewo ma za zadanie poradzić sobie z błędami poprzedniego drzewa. Do boosting'u w funkcji `gbm` dorzucamy walidację krzyżową, żeby od razu można było porównywać modele i szukamy optymalnego współczynnika uczenia. Ostatecznie najniższą wartość RMSE równą 101 230 dostajemy dla **Model_13** z parametrem `Shrinkage` = 0.05.

3.10 Bayesian Additive Regression Trees (BART)

Użyjemy teraz algorytmu BART, który ma za zadanie iteracyjnie dopasować drzewa do reszt, co pozwala skutecznie uchwycić złożone wzorce w danych. Użyliśmy innej funkcji niż na zajęciach ponieważ nie obsługiwała ona zmiennych typu factor, a także źle działała dla walidacji krzyżowej. Funkcja `bart` użyta przez nas pochodzi z biblioteki `dbarts`. **Model_14** oparty na tej funkcji uzyskał najniższy błąd RMSE = 99 896 dla tej funkcji z domyślnymi parametrami.

3.11 XGBoost

Ostatnią metodą użytą przez nas jest XGBoost, która do poprawy jakości modelu wykorzystuje spadek wzdłuż gradientu, co pozwala tworzyć drzewa decyzyjne w taki sposób, aby zminimalizować błędy predykcji w kolejnych iteracjach. Wykorzystaliśmy funkcję `xgb.cv` do oceny modelu za pomocą walidacji krzyżowej. W celu znalezienia optymalnego modelu sprawdziliśmy wiele modeli z różnymi parametrami: `max_depth`, `eta`, `nrounds`. Najlepszy okazał się model z następującymi parametrami:
`max_depth` = 3, `eta` = 0.01, `nrounds` = 1000. Końcowy **Model_15** osiągnął RMSE równe 100 129. (*Rysunek 13*)



Rysunek 13: Błąd RMSE a iteracja

4 Wybór najlepszego modelu

W poniższej tabeli (*Rysunek 14*) przedstawiliśmy wyniki błędów RMSE oraz użyte metody. Jak możemy zauważyć najlepsze wyniki uzyskały metody takie jak Random Forest, Boosting, BART oraz XGBoost, z czego najniższym błędem charakteryzuje się metoda BART - 99 896 RSME. W naszej opinii jest to nadal duży błąd, ponieważ większość domów kosztowała między 300 tys a 400 tys więc średni błąd wynosi około 1/3, 1/4 wartości całego domu. Jednak najprawdopodobniej z dostępnych danych nie da się uzyskać modeli lepszej jakości. Oprócz tego w naszej analizie można dostrzec, że wszystkie zmienne były istotne (może poza zmienną `bedrooms`) ale zmiennymi najbardziej istotnymi były `quartile_zone`, `living_in_m2`, `nice_view`, `grade` co wiemy między innymi z tego w jaki sposób były dokonywane podziały drzew.

Porównanie modeli

Model	RMSE	Metoda
Model_1	110081.46	GLM
Model_2	110011.82	GLM
Model_3	103551.49	GLM
Model_4	103586.29	GLM
Model_5	103569.14	Selekcja Krokowa
Model_6	104762.70	Regresja Grzbietowa
Model_7	104602.92	Regresja Lasso
Model_8	104785.21	PCR
Model_9	103832.97	PLS
Model_10	107041.14	Drzewo Regresyjne
Model_11	107513.23	Bagging
Model_12	100698.60	Random Forest
Model_13	101230.71	Boosting
Model_14	99896.59	BART
Model_15	100129.67	XGBoost

Rysunek 14: Porównanie modeli