Analiza metod ensemble learning

Zaawansowane Metody Uczenia Maszynowego

Grupa 1

Wydział Matematyki i Nauk Informacyjnych

9 czerwca 2025

Plan prezentacji

- Wstęp
  - Cel projektu
  - Zbiory danych
- Modele
  - Rozważane modele
  - Dobór parametrów
- Wyniki

Wstęp

Współczesne problemy uczenia maszynowego charakteryzują się rosnącą złożonością oraz obecnością dużych i często niejednorodnych zbiorów danych. W takich warunkach pojedyncze modele predykcyjne mogą nie być wystarczające.

Ensemble learning, to technika polegająca na łączeniu kilku różnych modeli w celu uzyskania lepszej jakości predykcji niż w przypadku pojedynczego modelu. Metody ensemble mogą poprawiać zarówno dokładność predykcji, jak i ogólną odporność modelu na przeuczenie.

Różne metody *ensemble learning* są powszechnie stosowane w systemach uczenia maszynowego. Mimo ich popularności, wybór odpowiedniej i jej implementacja często opiera się na intuicji lub doświadczeniu, a nie na dogłębnej analizie.

Cel projektu

Celem projektu jest porównanie skuteczności trzech najpopularniejszych technik ensemble learning (bagging, boosting, stacking) w zadaniach regresyjnych oraz analiza wpływu ich parametrów na odporność modelu na overfitting.

Motywacją do realizacji projektu jest chęć dokładnego zbadania i zrozumienia, w jakich warunkach warto stosować dane metody w kontekście rzeczywistych zadań.

Postawione pytania badawcze

- Które podejście zapewnia najniższy błąd predykcyjny dla każdego zestawu danych?
- Czy istnieją spójne wzorce, które mogą informować o przyszłym wyborze modelu?

Zbiory danych

W celu przeprowadzenia eksperymentów wybrano sześć różnych zbiorów regresyjnych, zróżnicowanych pod względem liczby obserwacji, cech oraz stopnia nieliniowości zależności. Do analizy wykorzystano zarówno dane rzeczywiste, jak i syntetyczne, co pozwala na szersze porównanie skuteczności modeli.

Przed przystąpieniem do eksperymentów wszystkie zbiory danych zostały poddane procesowi wstępnego przetwarzania (preprocessingu).

Airfoil Self Noise

Zbiór zawierający pomiary dotyczące hałasu emitowanego przez skrzydła samolotu w zależności od parametrów przepływu powietrza.

Liczba obserwacji: 1503

Liczba cech: 5 oryginalnych, 3 dodane

f, alpha, c, U_infinity, delta

f * delta, delta^2, $\textit{alpha}^2$

Zmienna objaśniana:
SSPL (poziom hałasu)

Źródło: UCI Machine Learning Repository

Przeprowadzono redukcję wymiarowości za pomocą analizy głównych składowych (PCA):
(alpha, delta) $\rightarrow$ alpha_delta_pc1.

Zastosowano transformację logarytmiczną log1p dla zmiennych:
f oraz delta.

Przeprowadzono one-hot encoding dla zmiennej c.

California Housing

Zbiór zawierający informacje o warunkach mieszkaniowych w Kalifornii. Jego celem jest przewidywanie średniej ceny domów.

Liczba obserwacji: 20640

Liczba cech: 8 oryginalnych, 4 dodane

MedInc, HouseAge, AveRooms, AveBedrms, Population, AveOccup, Latitude, Longitude

MedInc^2,
HouseholdDensity (Population / AveOccup),
MedInc * RoomsPC

Dyskretyzacja zmiennych:
HouseAge oraz MedInc

Zmienna objaśniana:
target (średnia wartość domów)

Źródło: UCI Machine Learning Repository

Przeprowadzono redukcję wymiarowości za pomocą analizy głównych składowych (PCA):
(Latitude, Longitude) $\rightarrow$ Geo1
(AveRooms, AveBedrms) $\rightarrow$ RoomsPC.

Zastosowano transformację logarytmiczną log1p dla zmiennych:
Population, AveRooms i AveBedrms.

Energy Efficiency

Zbiór pochodzi z badań nad efektywnością energetyczną budynków mieszkalnych. Celem jest przewidywanie dwóch zmiennych na podstawie cech związanych z konstrukcją i otoczeniem budynku.

Liczba obserwacji: 768

Liczba cech: 8 oryginalnych, 7 dodanych

Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, Glazing Area Distribution

Volume (Surface Area * Overall Height),
Surface_to_Volume_Ratio (Surface Area/Volume),
Wall_to_Roof_Ratio (Wall Area/Roof Area),
Height_to_Wall_Ratio (Overall Height/Wall Area),

Glazing_x_Orientation (Glazing Area * Orientation),
Glazing_x_GlazingDist (Glazing Area * Glazing Area Distribution),
Surface_x_Compactness (Surface Area * Relative Compactness),
Glazing_Area_Distribution_0 (Glazing_Area_Distribution == 0)

Zmienna objaśniana:
Heating Load, Cooling Load (zapotrzebowania na ogrzewanie oraz chłodzenie)

Usunięte zmienne:
Orientation, Glazing Area Distribution, Cooling Load

Źródło: UCI Machine Learning Repository

Friedman1

Syntetyczny zbiór danych generowany losowo zgodnie z funkcją wyrażoną wzorem
$y = 10 \cdot \sin(\pi x_1 x_2) + 20 \cdot (x_3 - 0.5)^2 + 10 \cdot x_4 + 5 \cdot x_5 + \varepsilon$,
gdzie x_i oznacza i-tą kolumnę macierzy X oraz $\varepsilon \sim \mathcal{N}(0, 1)$.

Liczba obserwacji: 5000

Liczba cech: 20 oryginalnych, 3 dodane

feature_0, feature_1, . . . , feature_19

feature_0 * feature_1
(feature_2)^2,
(feature_2) ^{2} – *feature_2*

Zmienna objaśniana: target

Usunięte zmienne:
feature_2, feature_5, . . . , feature_19

Źródło: wygenerowny z sklearn

Friedman3

Syntetyczny zbiór danych generowany losowo zgodnie z funkcją wyrażoną wzorem $y = \tan^{-1}\left(\frac{x_1}{x_2}\right) + \varepsilon$, gdzie x_i oznacza i-tą kolumnę macierzy X oraz $\varepsilon \sim \mathcal{N}(0, 0.1)$.
gdzie x_i oznacza i-tą kolumnę macierzy X oraz $\varepsilon \sim \mathcal{N}(0, 0.1)$.

Liczba obserwacji: 200

Liczba cech: 4 oryginalnych, 7 dodanych

feature_0, feature_1, feature_2, feature_3

feature_1 * feature_2,
(feature_1)^2,
(feature_2)^2,
feature_2_bin,
oraz trzy cechy binarne: feature_2_low,
feature_2_med,
feature_2_high

Zmienna objaśniana: target

Przeprowadzono dyskretyzację zmiennej feature_2 uzyskując feature_2_bin. Dodatkowo przeprowadzono one-hot encoding dla zmiennej feature_2_bin (feature_2_low, feature_2_med, feature_2_high).

Źródło: wygenerowny z sklearn

Regression

Zbiór regresyjny wygenerowany losowo.

Liczba obserwacji: 1000

Liczba cech: 10 oryginalnych, 0 dodanych

feature_0, feature_1, . . . , feature_9

Zmienna objaśniana: target

Usunięte zmienne:
feature_9

Źródło: wygenerowny z sklearn

Modele rozważane w projekcie

- **Bagging** $\rightarrow$ Random Forest Regressor
- **Boosting** $\rightarrow$ XGBoost Regressor
- **Stacking** $\rightarrow$ Ridge Regression (meta-learner)
  - wejście: predykcje RF, XGB + predykcja modelu OLS

Wybór najlepszych parametrów

W celu dobrania najlepszych parametrów dla każdego zbioru i dla każdego z modeli (RF i XGB) używamy Optuny. Jest to platforma do optymalizacji hiperparametrów typu open source, służąca do automatyzacji wyszukiwania hiperparametrów.

Struktura

- **Objective**: jedna funkcja, w której:
  1. Deklaracja przestrzeni możliwych wartości parametrów.
  2. Budowa modelu i zdefiniowanie funkcji celu (w naszym przypadku metryka RMSE, którą minimalizujemy).
- **Sampler**: `TPESampler(seed)`.
- **Direction**: im mniejszy RMSE, tym lepiej.

Sampler w Optunie

Punkt (θ) konkretna konfiguracja hiperparametrów (wektor liczb/kategorii).

Próba (trial) jedno uruchomienie `objective()` z punktem $\theta \rightarrow$ zwraca metrykę.

Sampler algorytm generujący kolejne punkty do oceny (u nas TPE).

Dlaczego TPE bije grid i random search?

- Koncentruje próby w „obiecujących" rejonach $\rightarrow$ mniej wywołań `objective()` do osiągnięcia tego samego RMSE.
- Obsługuje warunkowe (drzewiaste) przestrzenie i zmienne ciągłe, dyskretne, kategoryczne bez ręcznego dzielenia siatki.
- Skaluje się logarytmicznie z liczbą wymiarów, podczas gdy grid eksploduje kombinatorycznie.

TPESampler – algorytm krok po kroku

1. Start: Losowanie punktów z równym prawdopodobieństwem aż zapełniony zostanie bufor `n_startup_trials`.
2. Podział: Podział dotychczasowych wyników na dwie grupy
   - „dobre": $y < q$ (kwantyl) $\Rightarrow$ $l(\theta)$
   - „słabe": reszta $\Rightarrow$ $g(\theta)$
3. Modele estymujące gęstości: Dla każdego parametru, TPE uczy GMM $l(\theta)$ do zestawu wartości parametrów powiązanych z najlepszymi wartościami celu, oraz inny GMM $g(\theta)$ do pozostałych wartości parametrów.
4. Wybór następnego kroku: Wybierana jest nowa konfiguracja θ^{t+1} maksymalizująca stosunek $\frac{l(\theta)}{g(\theta)}$ (wysoka oczekiwana poprawa).
5. Aktualizacja: Trenowanie modelu, zapis RMSE w `study.trials`, powrót do kroku 2 aż do `n_trials`/`timeout`.

Hiperaparametry poddane strojeniu

Random Forest

- n_estimators
- max_depth
- min_samples_split
- min_samples_leaf
- max_features
- bootstrap
- max_samples
- criterion
- ccp_alpha

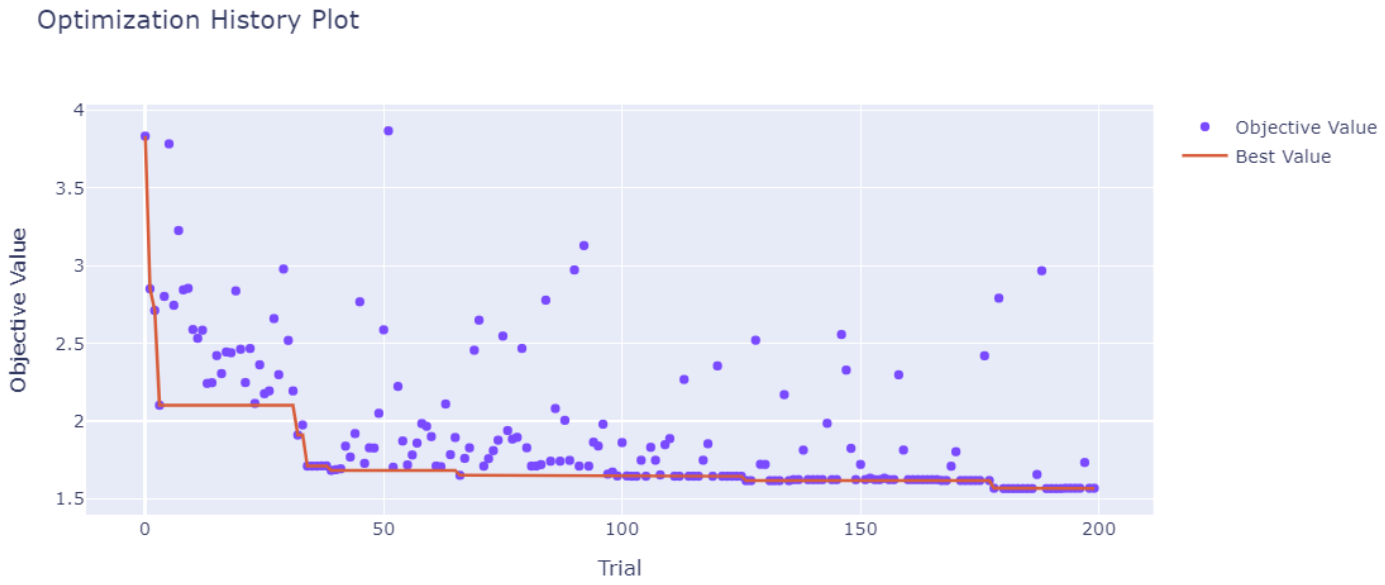
Ridge (meta-learner)

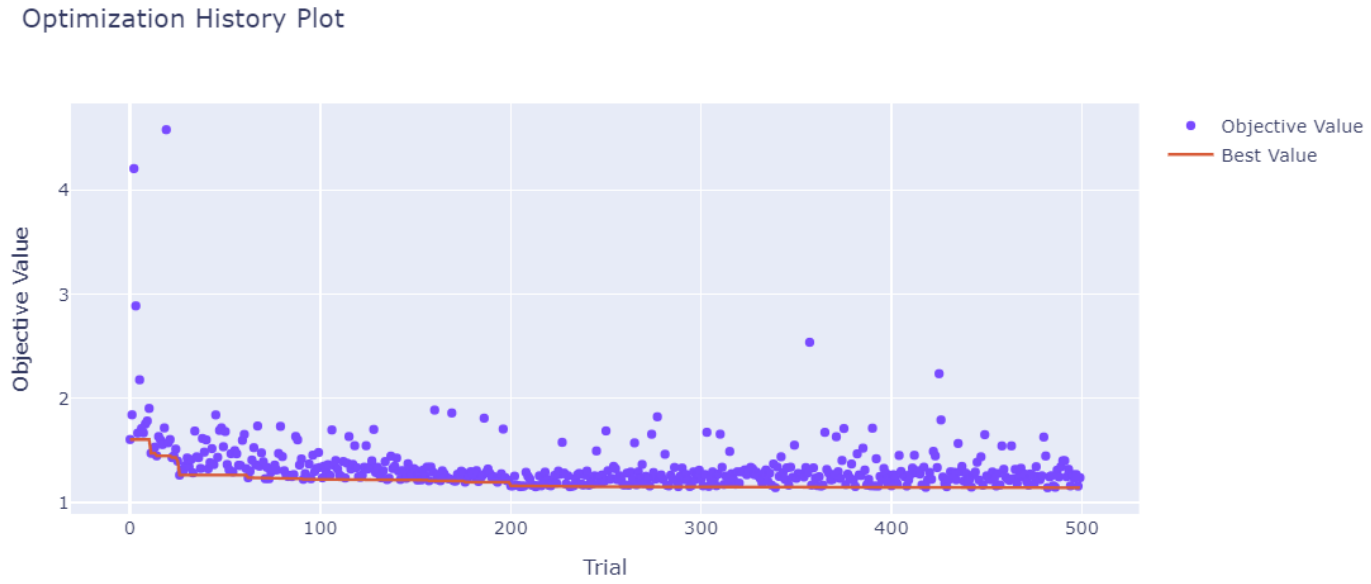
- alpha

XGBoost

- n_estimators
- learning_rate
- max_depth / max_leaves
- min_child_weight
- subsample
- colsample_bytree
- gamma
- reg_lambda
- reg_alpha
- max_delta_step
- grow_policy

Wykresy optymalizacji - Airfoil Self Noise



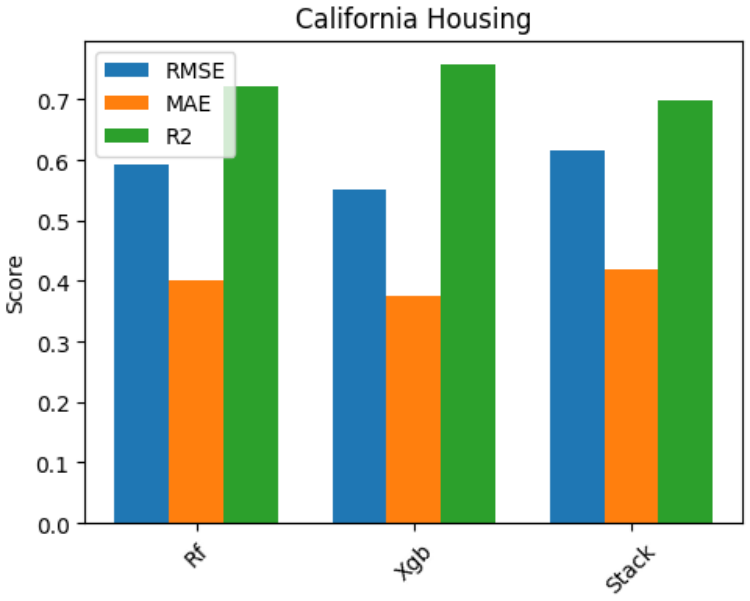


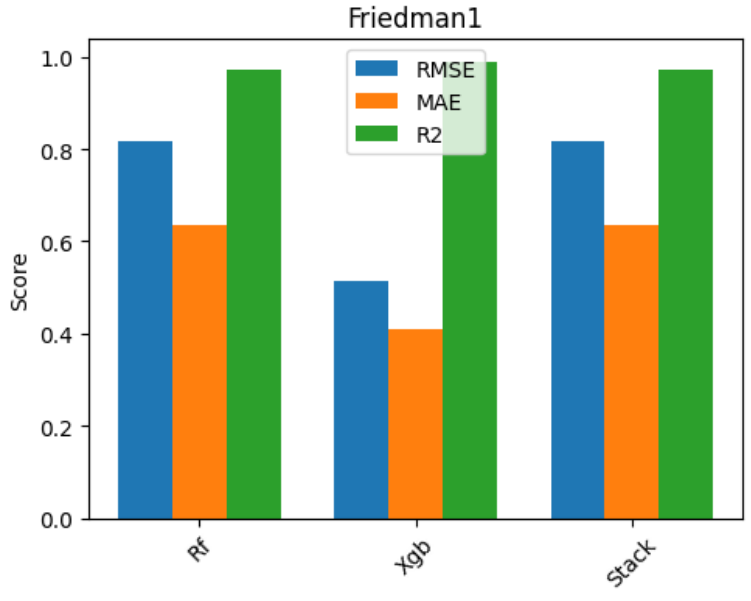
Wyniki

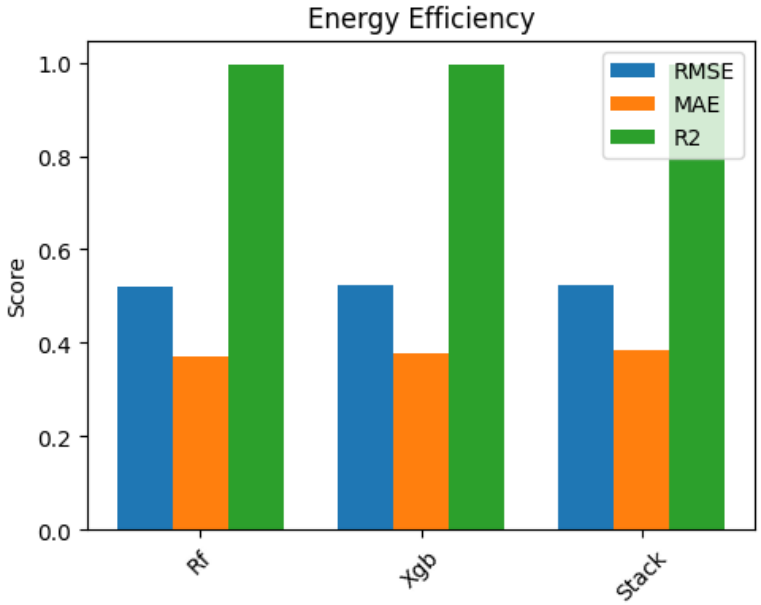
Porównania modeli dokonano opierając się o kilka kluczowych aspektów

- Oceniono dokładność predykcji modeli na zbiorze testowym, wykorzystując metryki RMSE, MAE oraz współczynnik determinacji R^2;
- Dla każdego modelu zbadano wartości odchylenia standardowego dla RMSE, MAE i R^2 w celu oceny stabilności modelu;
- Przeanalizowano różnice pomiędzy wynikami na zbiorze treningowym i testowym, aby ocenić odporność modeli na przeuczenie;
- Zbadano rozkład reszt dla każdego modelu;
- Porównano wartości przewidywane z rzeczywistymi, wykorzystując wykresy rozrzutu do wizualnej oceny jakości predykcji;
- Przeanalizowano wpływ poszczególnych zmiennych na wynik predykcji.

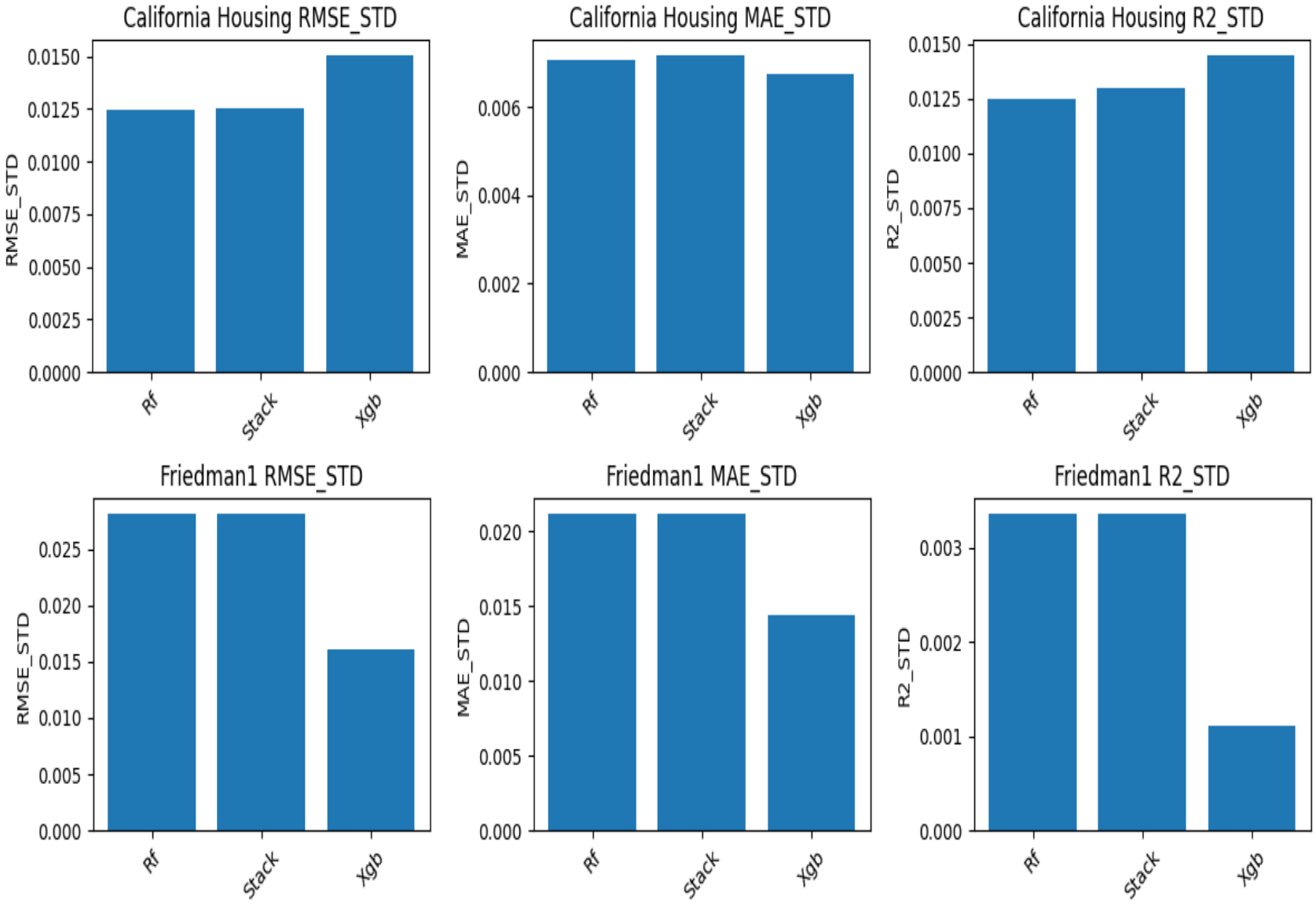
Porównanie metryk modeli na zbiorze testowym



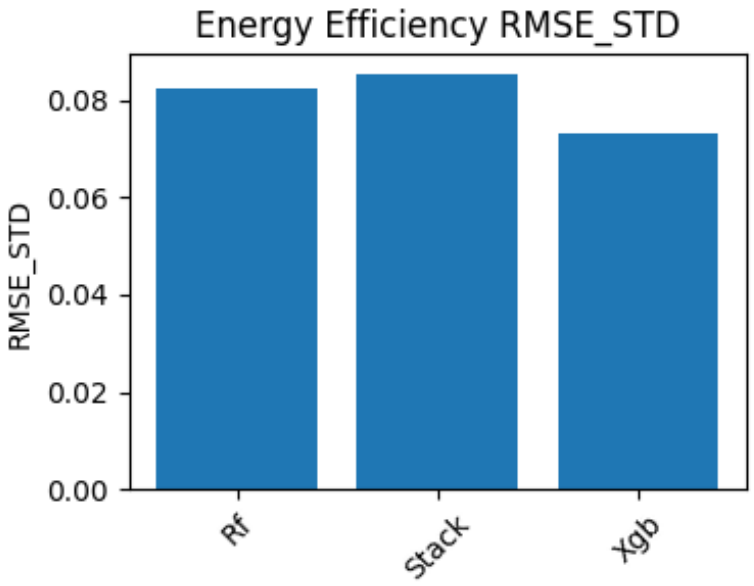


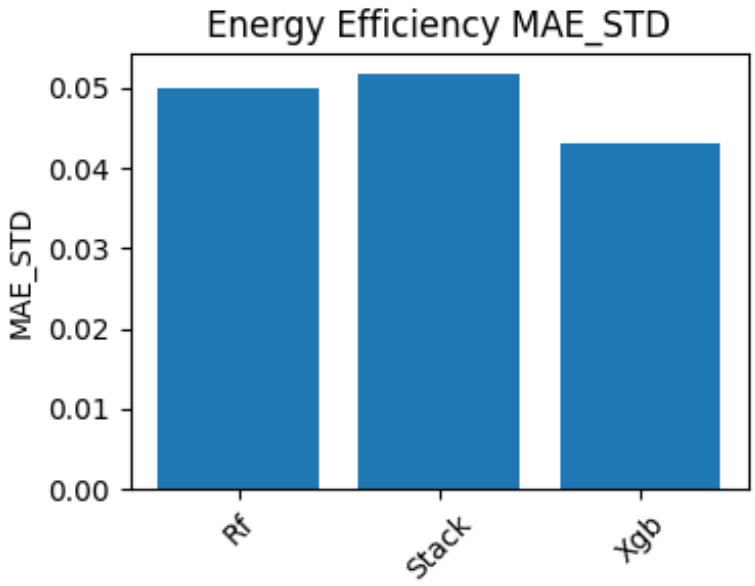


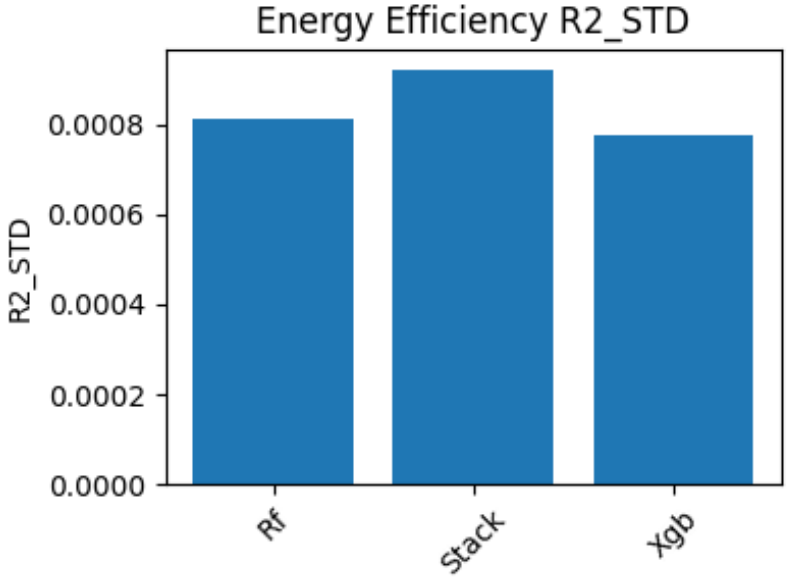
Wartości odchylenia standardowego dla każdego modelu



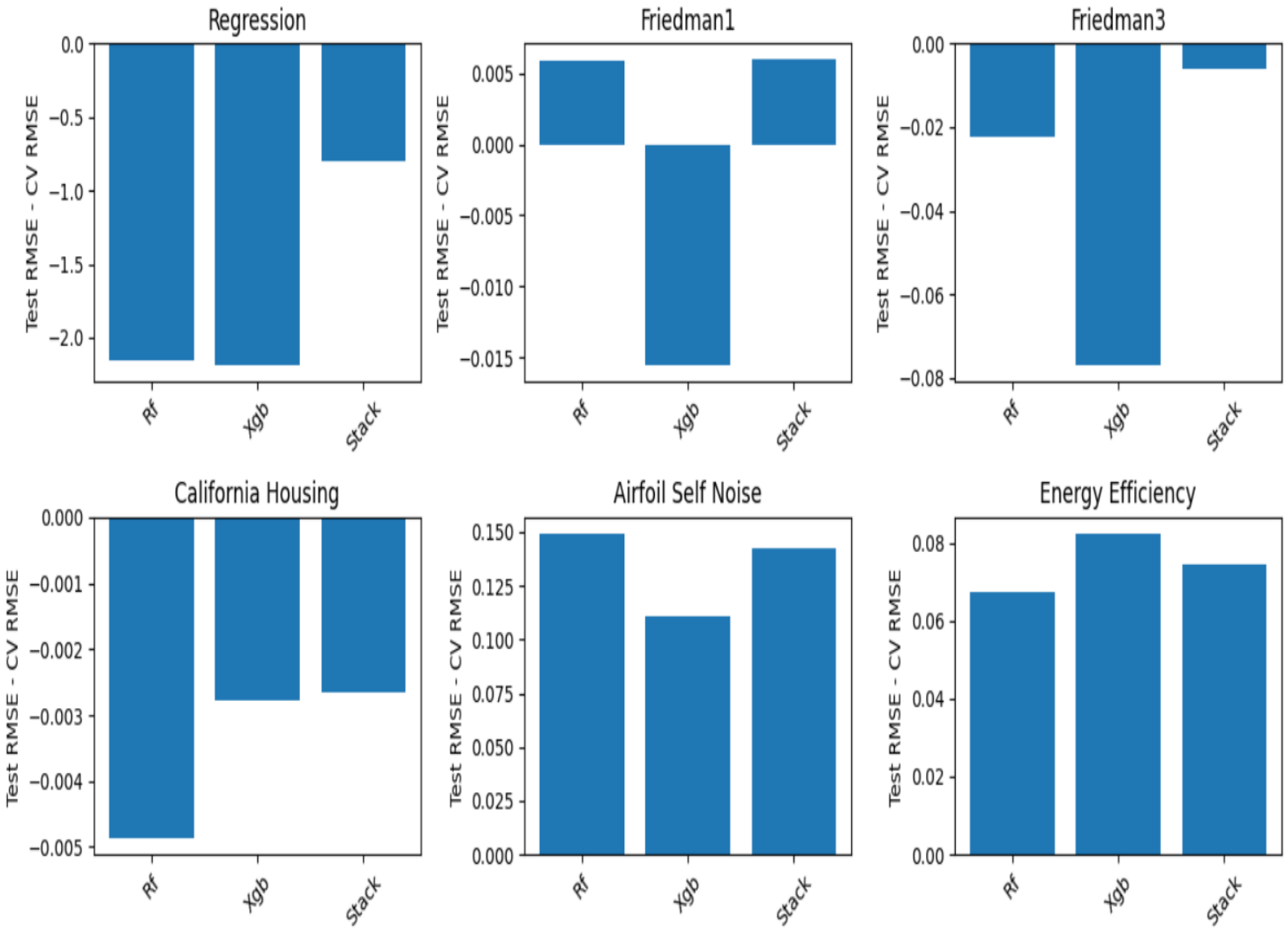
Wartości odchylenia standardowego - c.d.



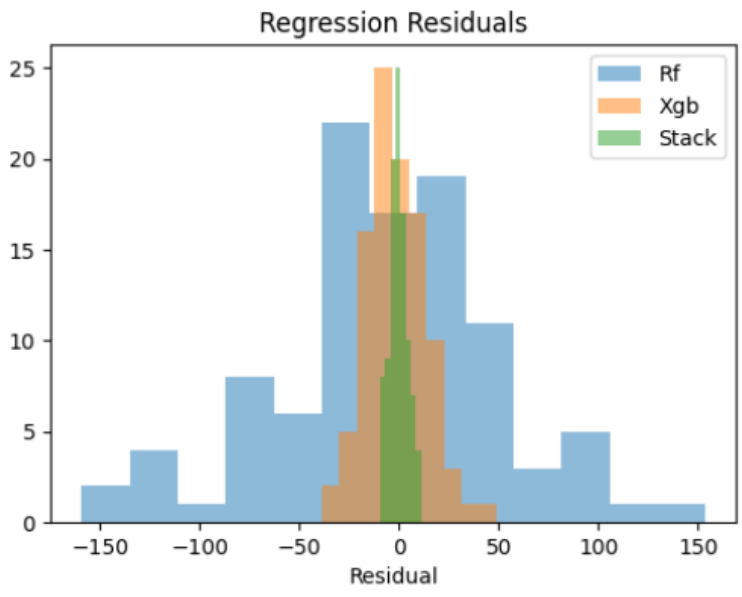


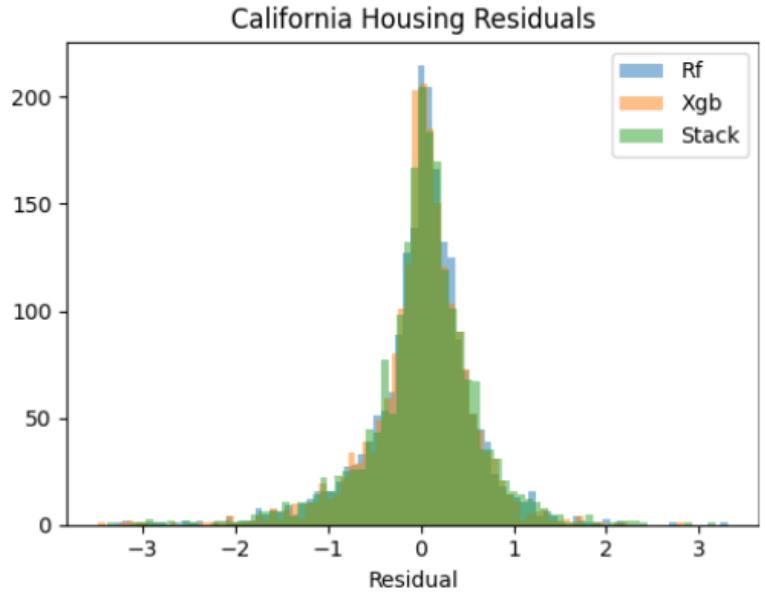


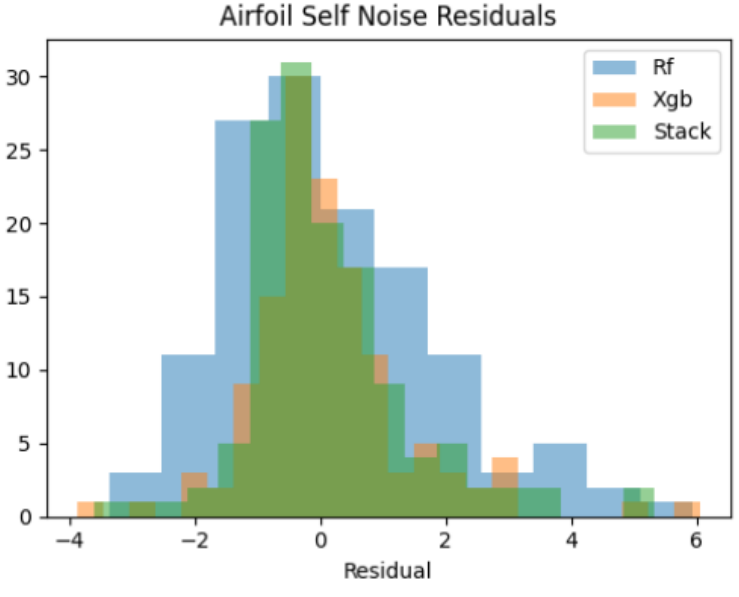
Wartości przeuczenia dla każdego modelu



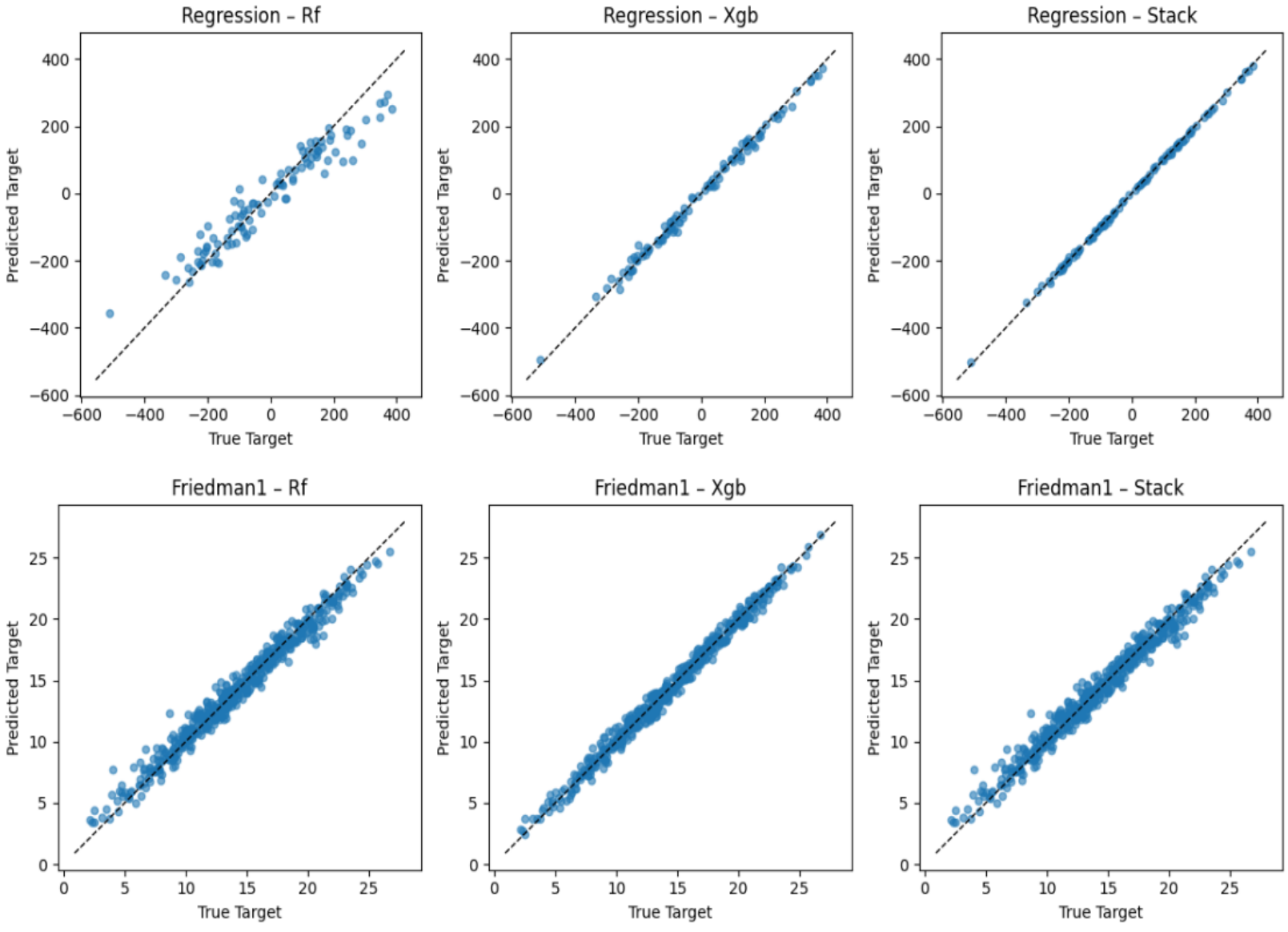
Rozkład błędów modeli

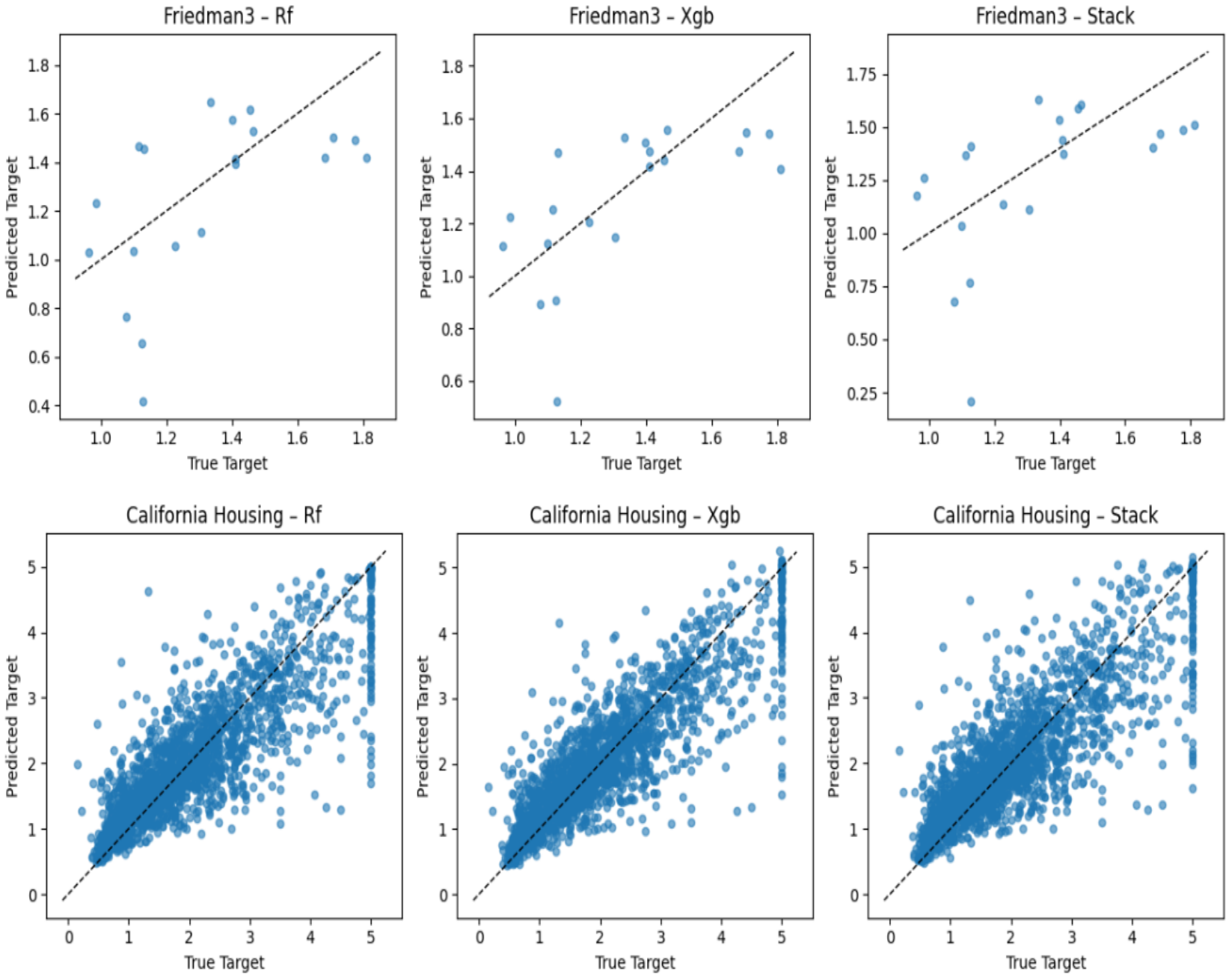


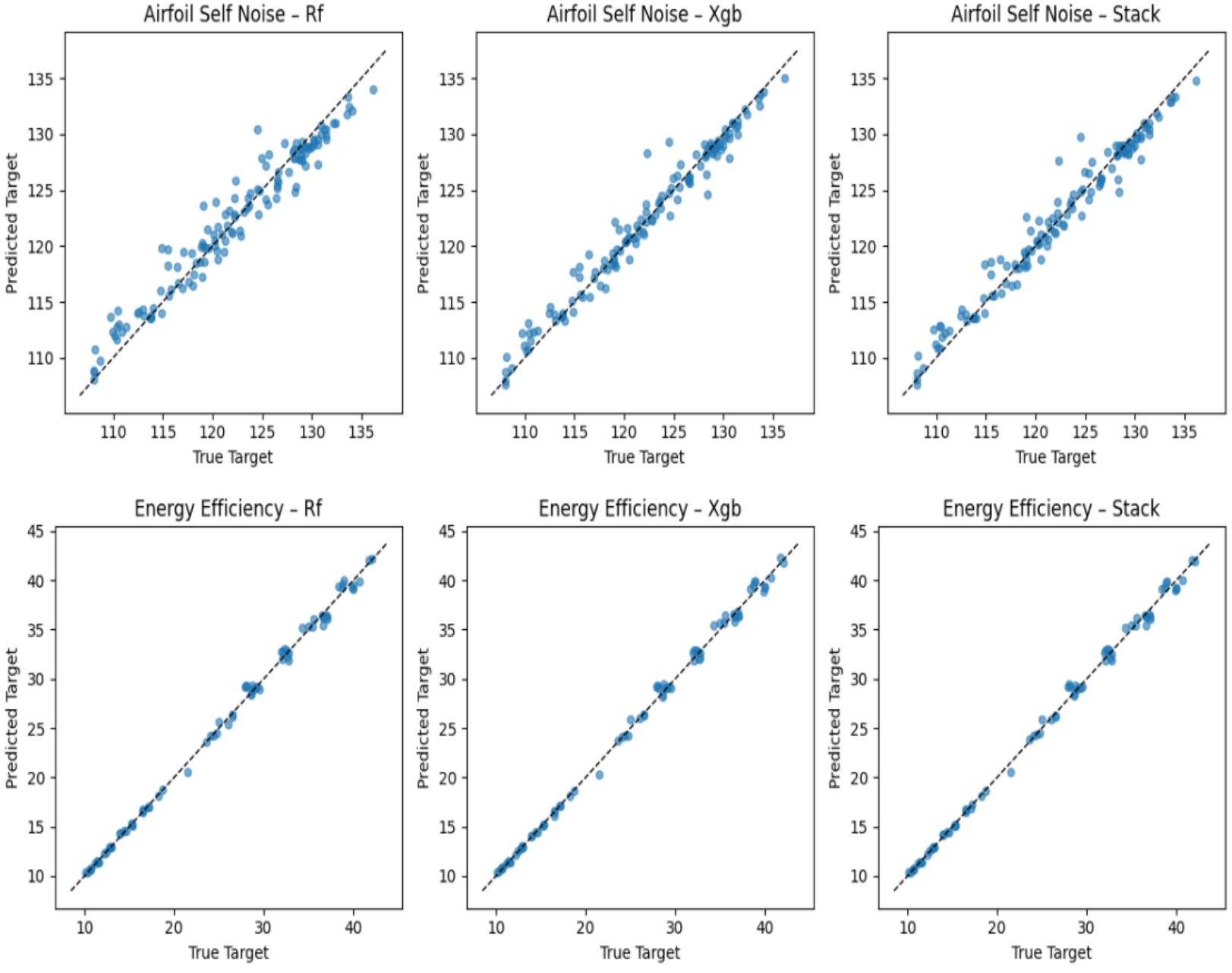




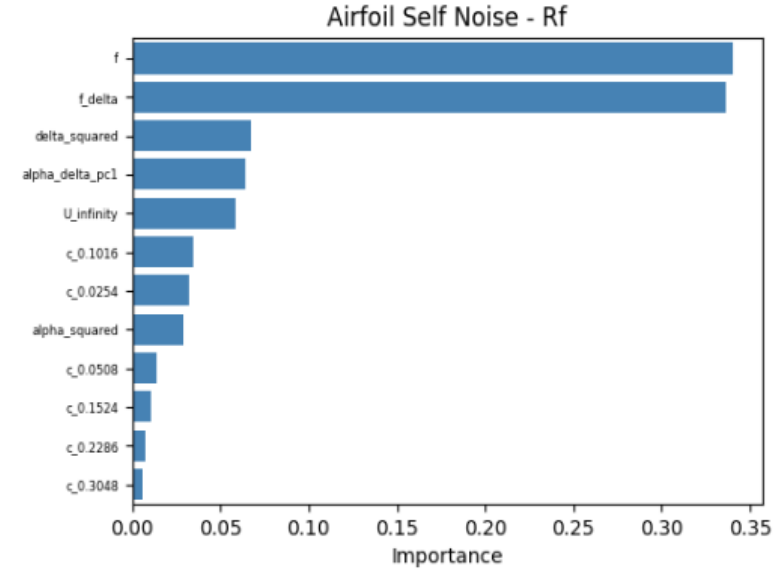
Wykresy rozrzutu wartości rzeczywistych i przewidywanych

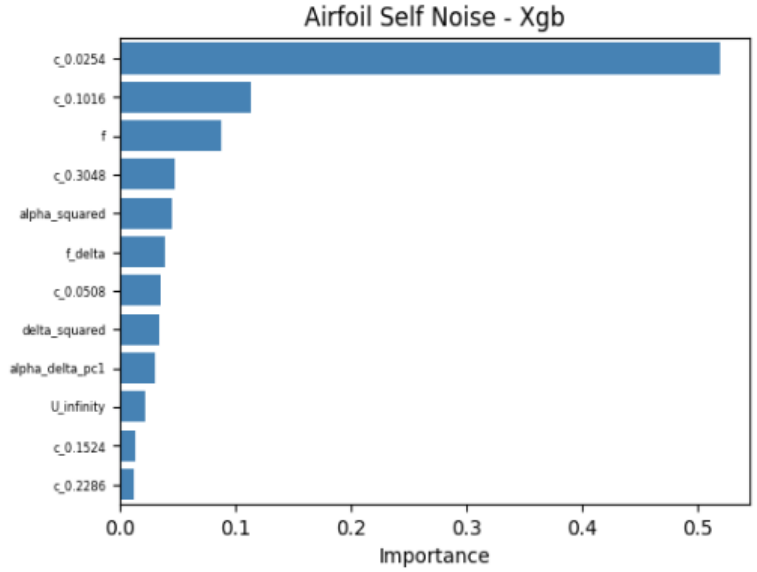


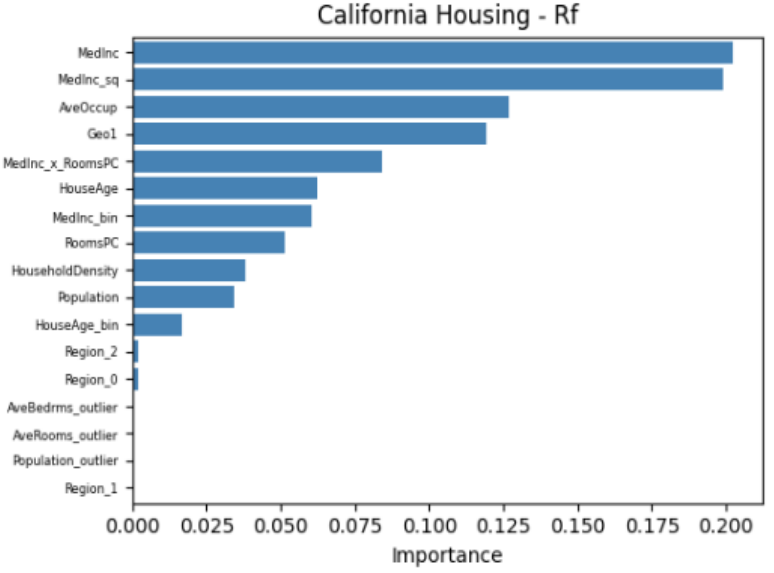

Friedman3 – Rf
Friedman3 – Xgb
Friedman3 – Stack
California Housing – Rf
California Housing – Xgb
California Housing – Stack
Predicted Target
True Target

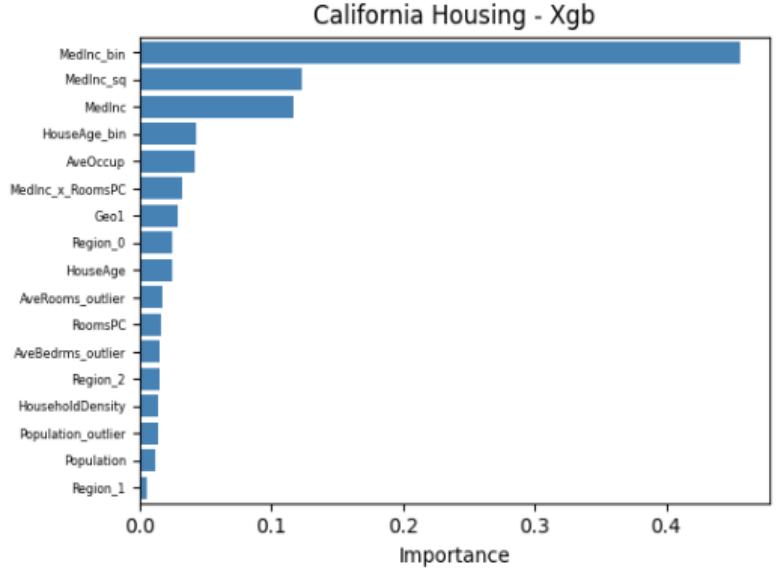

Airfoil Self Noise – Rf
Airfoil Self Noise – Xgb
Airfoil Self Noise – Stack
Energy Efficiency – Rf
Energy Efficiency – Xgb
Energy Efficiency – Stack
Predicted Target
True Target

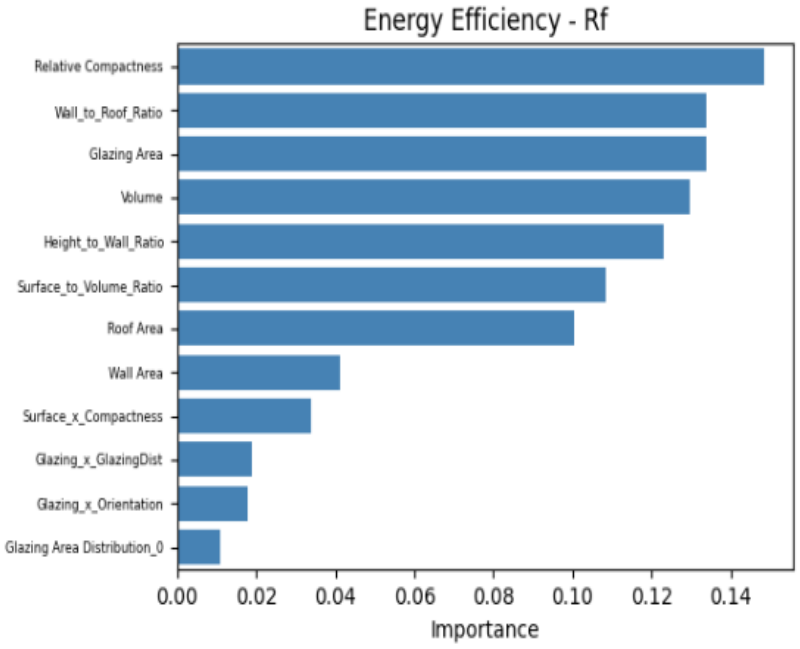
Istotność cech

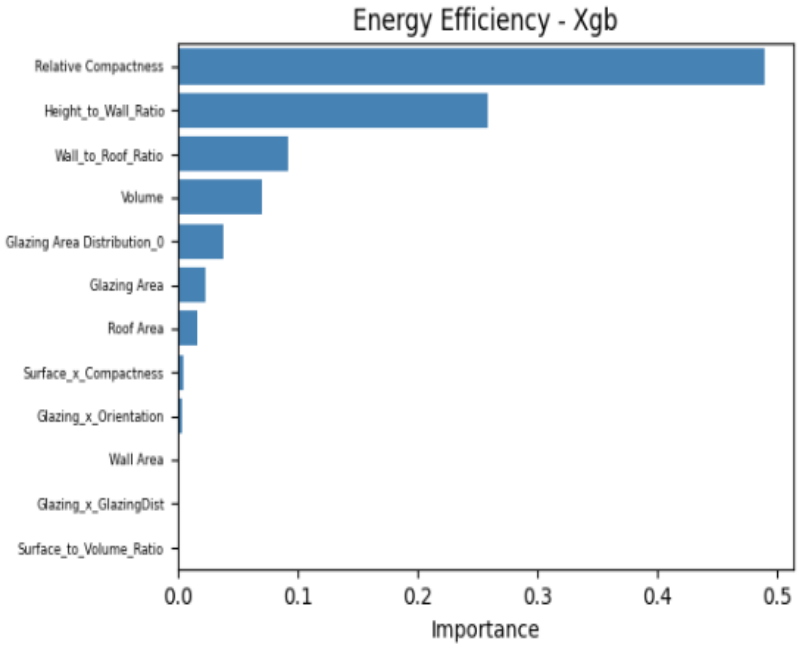


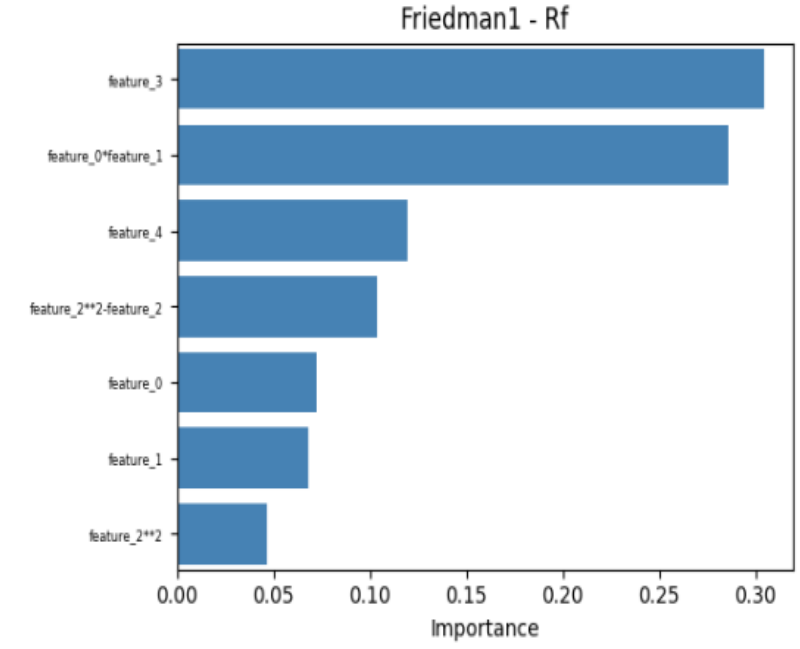


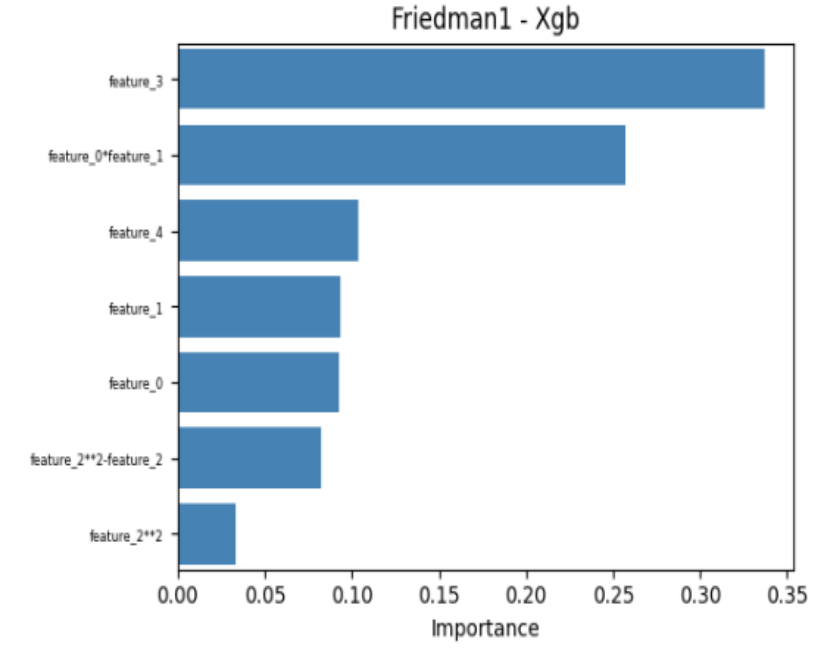


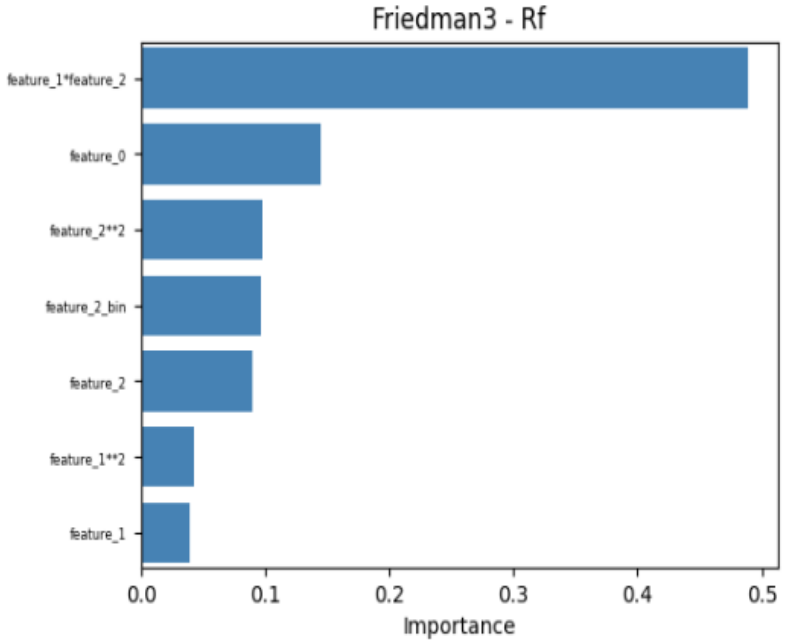


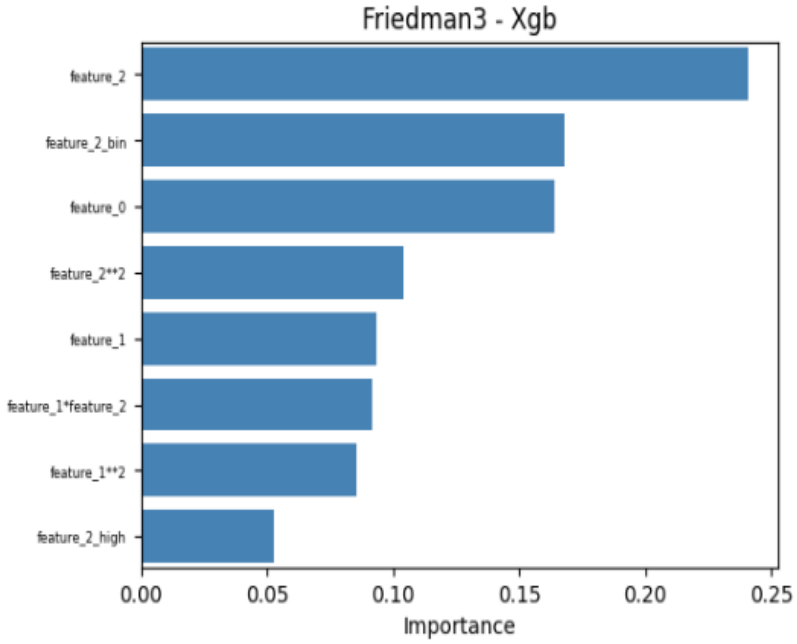

Energy Efficiency - Rf
Relative Compactness
Wall_to_Roof_Ratio
Glazing Area
Volume
Height_to_Wall_Ratio
Surface_to_Volume_Ratio
Roof Area
Wall Area
Surface_x_Compactness
Glazing_x_GlazingDist
Glazing_x_Orientation
Glazing Area Distribution_0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Importance

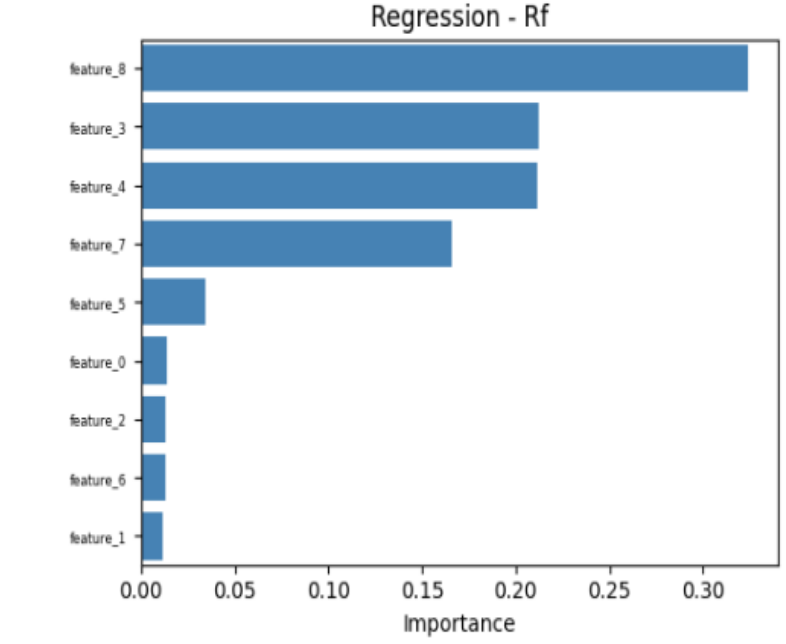

Energy Efficiency - Xgb
Relative Compactness
Height_to_Wall_Ratio
Wall_to_Roof_Ratio
Volume
Glazing Area Distribution_0
Glazing Area
Roof Area
Surface_x_Compactness
Glazing_x_Orientation
Wall Area
Glazing_x_GlazingDist
Surface_to_Volume_Ratio
0.0
0.1
0.2
0.3
0.4
0.5
Importance

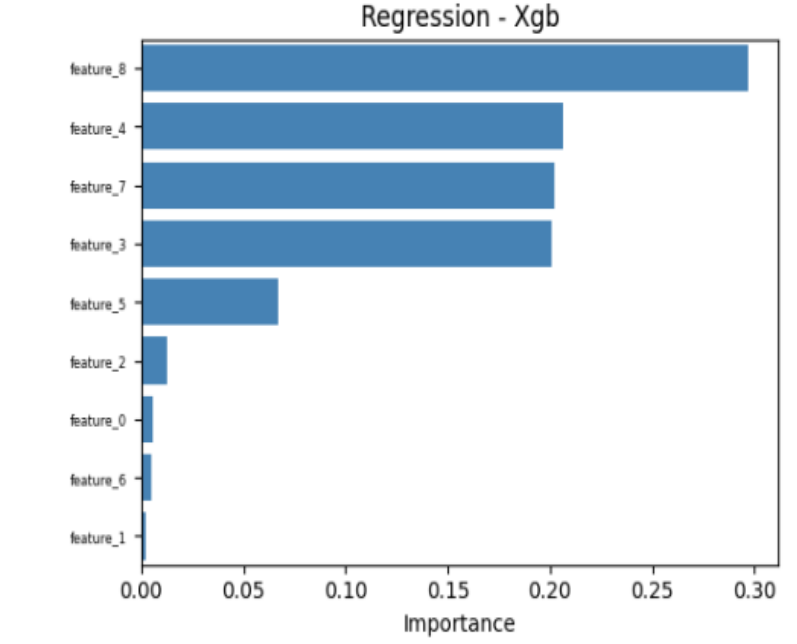

Friedman1 - Rf
feature_3
feature_0*feature_1
feature_4
feature_2**2-feature_2
feature_0
feature_1
feature_2**2
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance


Friedman1 - Xgb
feature_3
feature_0*feature_1
feature_4
feature_1
feature_0
feature_2**2-feature_2
feature_2**2
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance


Friedman3 - Rf
feature_1*feature_2
feature_0
feature_2**2
feature_2_bin
feature_2
feature_1**2
feature_1
0.0
0.1
0.2
0.3
0.4
0.5
Importance


Friedman3 - Xgb
feature_2
feature_2_bin
feature_0
feature_2**2
feature_1
feature_1*feature_2
feature_1**2
feature_2_high
0.00
0.05
0.10
0.15
0.20
0.25
Importance


Regression - Rf
feature_8
feature_3
feature_4
feature_7
feature_5
feature_0
feature_2
feature_6
feature_1
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance


Regression - Xgb
feature_8
feature_4
feature_7
feature_3
feature_5
feature_2
feature_0
feature_6
feature_1
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance

Podsumowanie wyników

Zbiór danych	Najlepszy model na zbiorze testowym*
regression (1k, synt.)	XGB (najniższy RMSE/MAE)
friedman1 (5k, synt.)	XGB
friedman3 (200, synt.)	XGB
california housing (20k, rzecz.)	XGB
airfoil self-noise (1.5k, rzecz.)	XGB
energy efficiency (768, rzecz.)	RF

*modele z najniższym RMSE (przy remisie wybór według MAE)

Podsumowanie modeli

Metoda	Wnioski
Boosting (XGB)	Najbardziej konsekwentnie osiąga najlepsze wyniki, zwłaszcza na większych lub złożonych danych
Bagging (RF)	Szybki i stabilny model; może wygrywać na uporządkowanych, średnich zbiorach danych
Stacking	Rzadko przewyższa XGB; główną zaletą jest lekko większy R^2 kosztem czasu i złożoności

Analiza metod ensemble learning

Zaawansowane Metody Uczenia Maszynowego

Grupa 1

Wydział Matematyki i Nauk Informacyjnych

9 czerwca 2025