

Raport z pracy domowej 1

Łukasz Tomaszewski

1 Wstęp

Celem pracy domowej było przeanalizowanie tunowalności trzech algorytmów uczenia maszynowego. Do realizacji tego zadania zostały przeze mnie wybrane cztery zbiory danych, dotyczące zadania regresji:

- [Abalone](#) (4177 x 9)
- [Auto-MPG](#) (398 x 9)
- [Insurance](#) (1338 x 7)
- [Concrete Compressive Strength](#) (1030 x 9).

Zbiory te zostały poddane podstawowemu preprocessingowi. Polegał on na zakodowaniu zmiennych kategorycznych oraz usunięciu wierszy, zawierających braki danych. Na tak przygotowanych zbiorach wytrenowane zostały trzy algorytmy z pakietu scikit-learn: Random Forest Regressor, Gradient Boosting Regressor oraz Elastic Net. Dla ostatniego algorytmu zmienne objaśniające zostały dodatkowo przeskalowane.

2 Wykorzystane metody

Dla każdego algorytmu zdefiniowana została przestrzeń, na której szukane były optymalne hiperparametry. Hiperparametry oraz ich zakresy wybrane zostały przy wykorzystaniu informacji z [1] oraz własnego doświadczenia.

Tabela 1: Przestrzeń przeszukiwań hiperparametrów

Algorytm	Hiperparametr	Typ	Ograniczenie dolne	Ograniczenie górne
Random Forest Regressor	n estimators	integer	1	1000
	max depth	integer	1	100
	min samples split	integer	2	10
	min samples leaf	integer	1	5
Gradient Boosting Regressor	n estimators	integer	1	1000
	learning rate	float	0.01	0.3
	subsample	float	0.1	1
	max depth	integer	1	100
	min samples split	integer	2	10
	min samples leaf	integer	1	5
Elastic Net	alpha	float	0	1
	l1 ratio	float	0	1

Dla każdego algorytmu, na każdym zbiorze uruchomiony został RandomSearchCV z pakietu scikit-learn oraz BayesSearchCV z pakietu scikit-optimize. Algorytmy te w każdym uruchomieniu wykonały po 200 iteracji (wybrały 200 punktów z przestrzeni przeszukiwań). Punkty wybrane przez Random Search były takie same dla wszystkich zbiorów. Dla każdej iteracji algorytmy wykonywały trzystopniową krosvalidację. Dla każdej krosvalidacji wyznaczona została miara R^2 , a ostatecznym wynikiem każdej iteracji była średnia z miar na trzech krosvalidacjach.

W celu wyznaczenia defaultowych hiperparametrów algorytmu, dla każdej siatki ewaluowanej przez Random Search mierzyliśmy średni wynik na wszystkich czterech zbiorach danych. Jako defaultowe hiperparametry wybraliśmy tę siatkę, która miała średnio najlepszy wynik na wszystkich zbiorach.

Tunowalność poszczególnych algorytmów wyznaczona została tak jak w artykule [1].

3 Wyniki

3.1 Porównanie wyników

Miarę R^2 w zależności od numeru iteracji przedstawiają wykresy na rysunku 2. Możemy zauważyć że dla algorytmów Random Forest Regressor oraz Gradient Boosting Regressor wykresy dla Random Search oraz Bayesian Search wyglądają podobnie. Inaczej sprawa ma się, gdy spojrzymy na wykresy algorytmu Elastic Net. Możemy zauważyć, że w przypadku Random Search wyniki są mocno rozrzucone w przeciwieństwie do Bayesian Search, gdzie wyniki skupione są wokół jednej wartości. Może to nam sugerować, że dobór odpowiednich hiperparametrów w przypadku tego algorytmu jest znacznie trudniejszy, niż w przypadku pozostałych dwóch. Podobne wnioski możemy wysnuć z rysunku 3. Warto zaznaczyć, że z rysunku 2 wynika także, że nie zawsze największe modele (wykonujące się najdłużej) są najlepsze.

Rysunek 4 zawiera wykresy, które przedstawiają najlepszy wynik do momentu danej iteracji. Możemy zaobserwować, że Bayesian Search o wiele szybciej znajduje najlepszą kombinację hiperparametrów, która w dodatku jest lepsza niż najlepsza kombinacja znaleziona przez Random Search. Istnieją jednak przypadki, w których Bayesian Search tkwi w minimum lokalnym, a Random Search znajduje lepszą konfigurację hiperparametrów.

3.2 Najlepsze hiperparametry i wyznaczenie defaultowych hiperparametrów

Tabela 2 zawiera najlepsze znalezione hiperparametry. Jak możemy zobaczyć hiperparametry te różnią się znacznie dla Random Search i Bayesian Search oraz dla różnych zbiorów danych.

Tabela 2: Najlepsze znalezione hiperparametry

		Abalone	Auto MPG	Insurance	Concrete
Random Forest Regressor					
n estimators	R	967	138	824	90
	B	441	941	817	150
max depth	R	8	69	4	15
	B	27	96	5	63
min samples split	R	2	3	5	2
	B	9	2	6	2
min samples leaf	R	3	1	4	1
	B	5	2	4	1
Gradient Boosting Regressor					
n estimators	R	22	527	22	412
	B	393	832	217	507
learning rate	R	0.17	0.15	0.17	0.12
	B	0.01	0.01	0.01	0.24
subsample	R	0.48	0.31	0.48	0.49
	B	0.1	0.38	0.19	0.70
max depth	R	3	48	3	1
	B	62	22	51	1
min samples split	R	9	7	9	5
	B	5	10	10	10
min samples leaf	R	3	3	3	4
	B	5	1	5	5
Elastic Net					
alpha	R	0.0026	0.1910	0.7049	0.0077
	B	0.0005	0.0	0.0827	0.1308
l1 ratio	R	0.98	0.96	0.99	0.08
	B	0.01	0.0	0.95	1.0

W tabeli 3 umieszczone zostały wyznaczone defaultowe hiperparametry. W tabeli 4 umieszczone zostały miejsca, które defaultowe hiperparametry zajęły w poszczególnych zbiorach danych. Możemy zobaczyć, że dla Random

Forest Regressor defaultowe hiperparametry były zarazem najlepsze oraz najslabsze w zależności od zbioru. Dla pozostałych algorytmów, miejsca te były do siebie zbliżone.

Tabela 3: Defaultowe hiperparametry

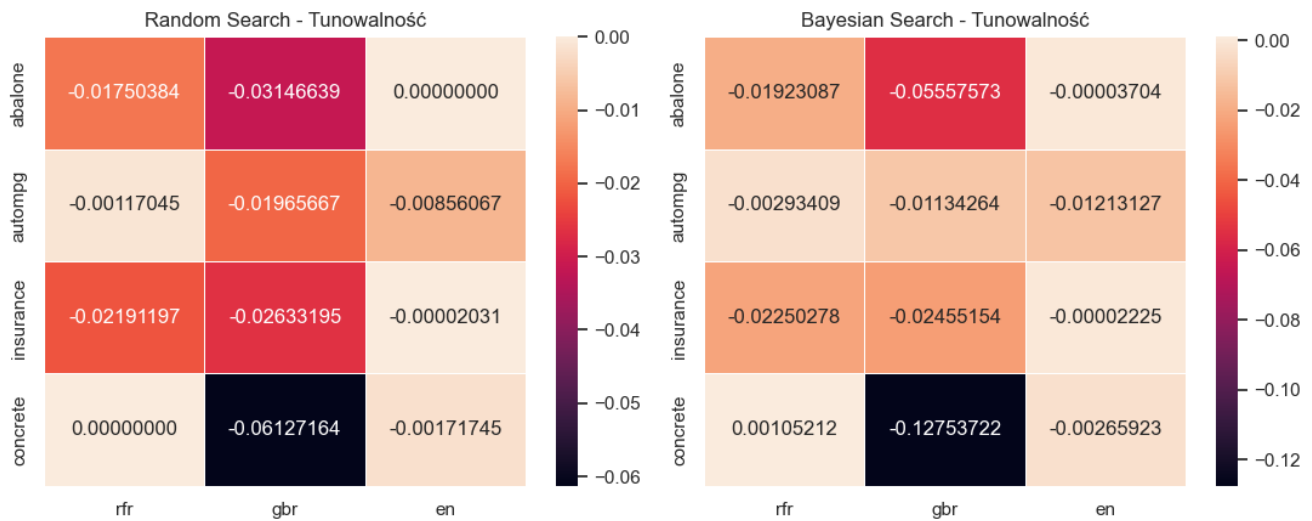
Algorytm	Hiperparametr	Wartość
Random Forest Regressor	n estimators	90
	max depth	15
	min samples split	2
	min samples leaf	1
Gradient Boosting Regressor	n estimators	839
	learning rate	0.0110
	subsample	0.27
	max depth	65
	min samples split	9
	min samples leaf	2
Elastic Net	alpha	0.0026
	l1 ratio	0.988

Tabela 4: Miejsca zajęte przez defaultowe hiperparametry

Zbiór	Random Forest	Gradient Boosting	Elastic Net
Abalone	178	6	1
Autompg	3	6	17
Insurance	192	6	14
Concrete	1	10	12

3.3 Tunowalność

Dla większości zbiorów tunowalność okazała się być ujemna, wyjątek stanowi zbiór *concrete* oraz *abalone* gdzie tunowalność wynosiła 0 w przypadku Random Search. Poprawę wyników dzięki użyciu innych hiperparametrów niż wyznaczone defaulty możemy zobaczyć na heatmapach 1. Dla obydwu sposobów przeszukiwania przestrzeni hiperparametrów poprawa była podobna.



Rysunek 1: Heat mapy przedstawiające tunowalność algorytmów

W obydwu sposobach losowania punktów największa poprawa wystąpiła w algorytmie Gradient Boosting Regressor na zbiorze *concrete* oraz *abalone*. Średnią poprawę wyników przy dostosowaniu hiperparametrów do zbioru

dla poszczególnych algorytmów możemy zobaczyć w tabeli 5.

Tabela 5: Średnia tunowalność

Metoda losowania hiperparametrów	Random Forest	Gradient Boosting	Elastic Net
Random Search	-0.010147	-0.034682	-0.002575
Bayesian Search	-0.010904	-0.054752	-0.003712

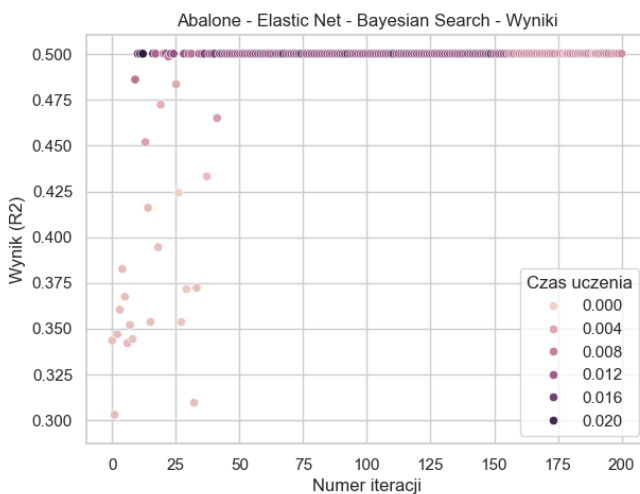
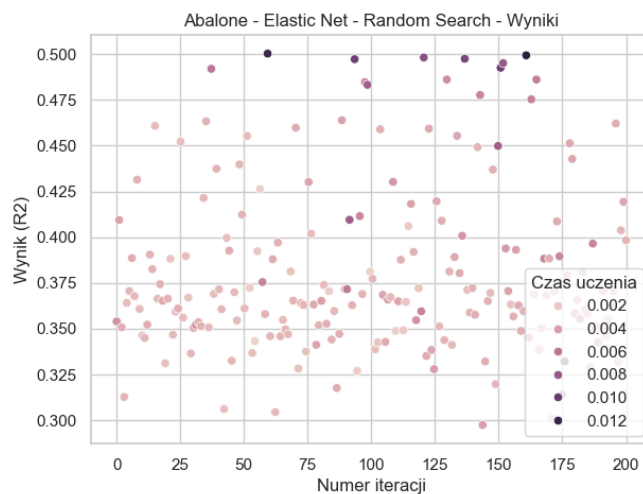
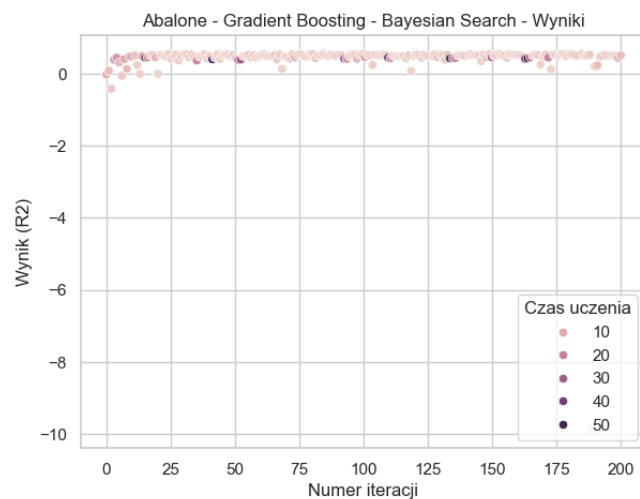
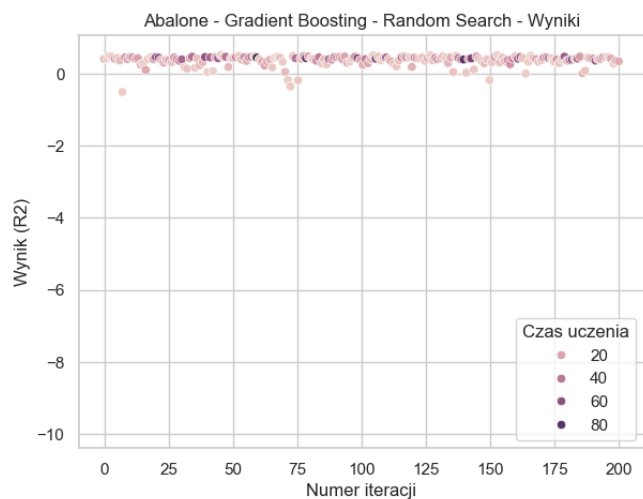
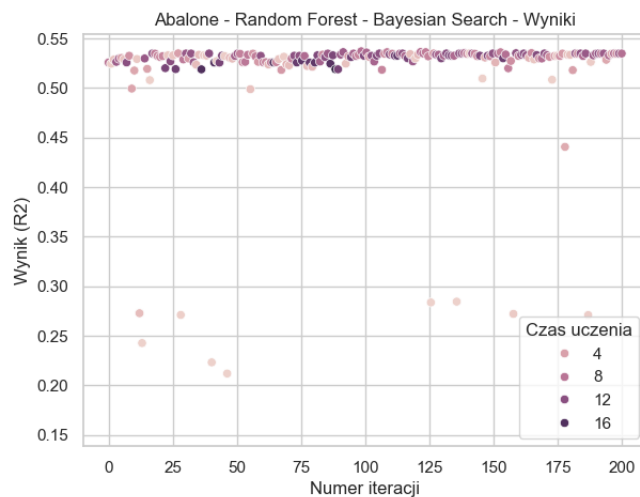
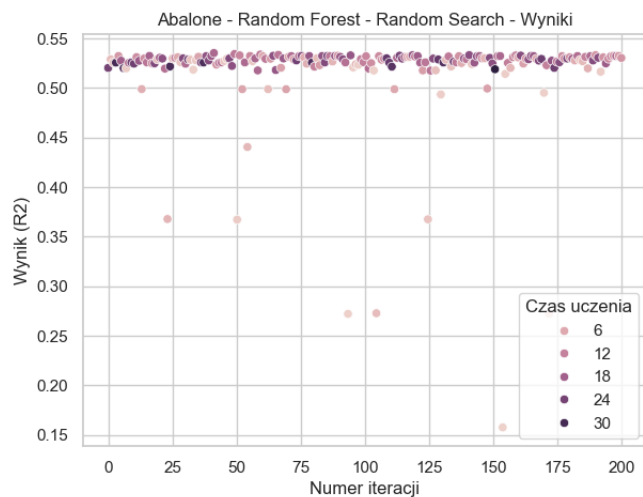
4 Podsumowanie

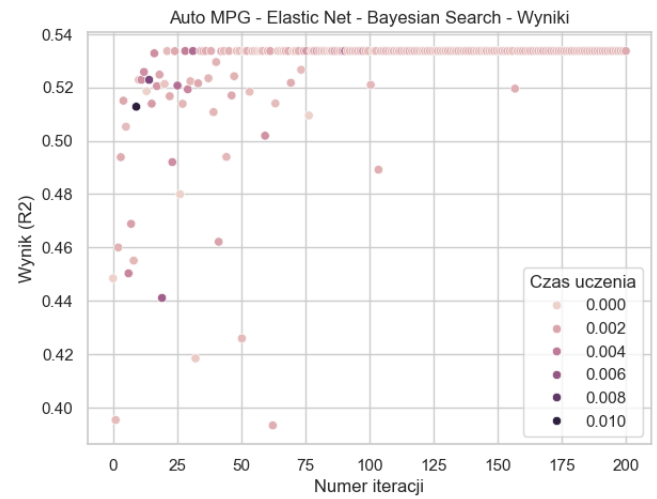
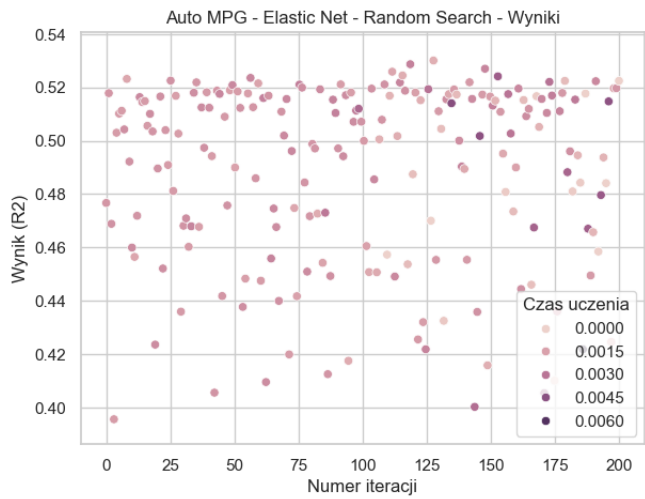
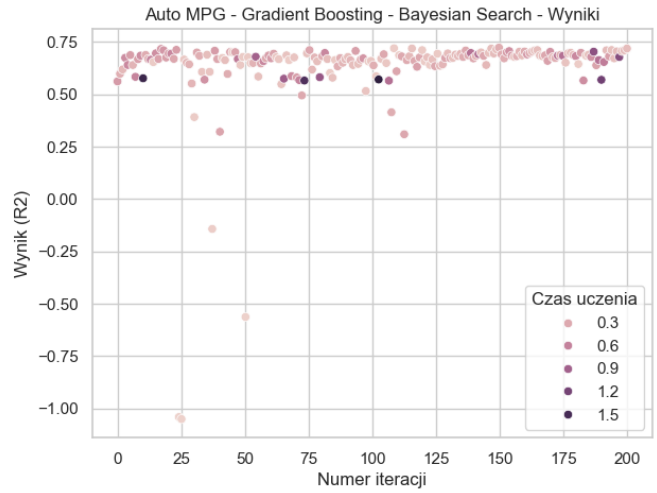
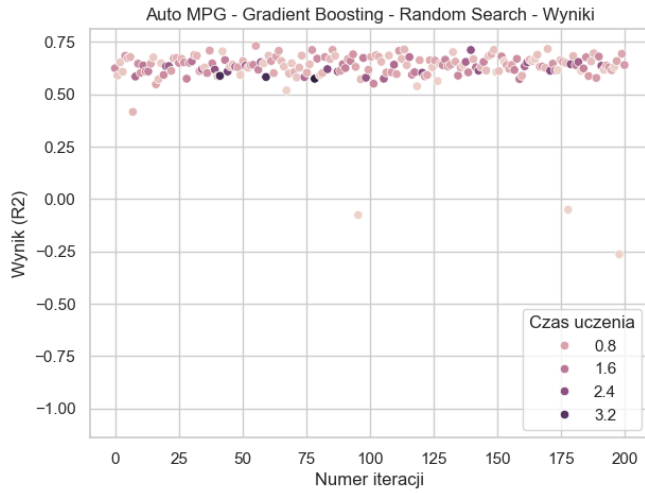
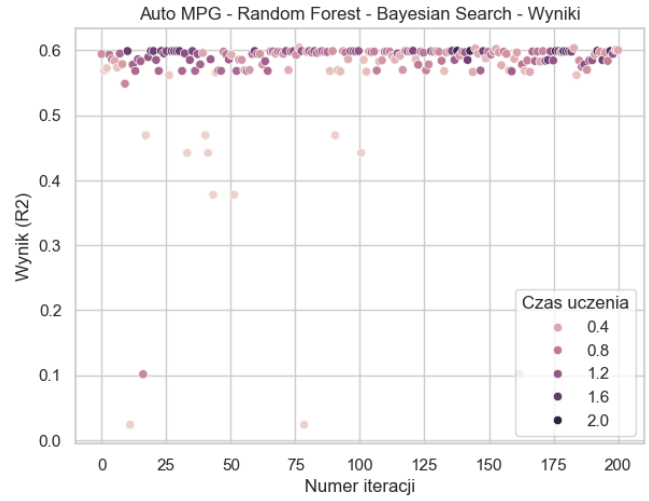
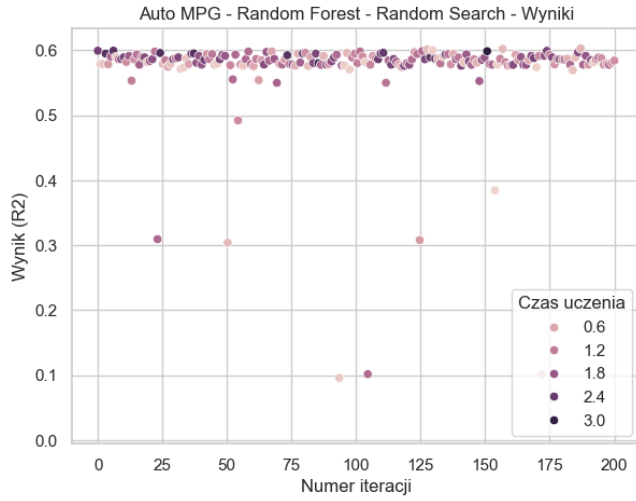
Z przeprowadzonych eksperymentów wynika, że wybrane algorytmy uczenia maszynowego osiągają lepsze wyniki, gdy dla każdego zbioru korzystamy z odpowiednio dobranych hiperparametrów, a nie korzystamy z tych defaultowych. Największą poprawę możemy zaobserwować dla algorytmu Gradient Boosting Regressor. Wnioski na temat tunowalności nie zależały od metody losowania hiperparametrów. Dla Random Search oraz Bayesian Search poprawa wyników dla poszczególnych algorytmów jest do siebie zbliżona.

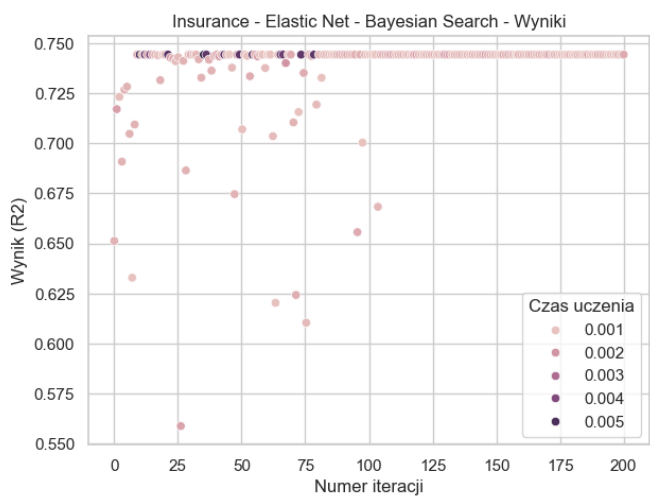
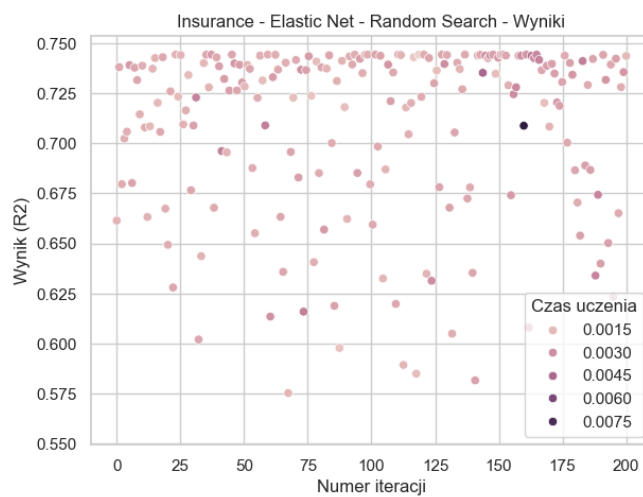
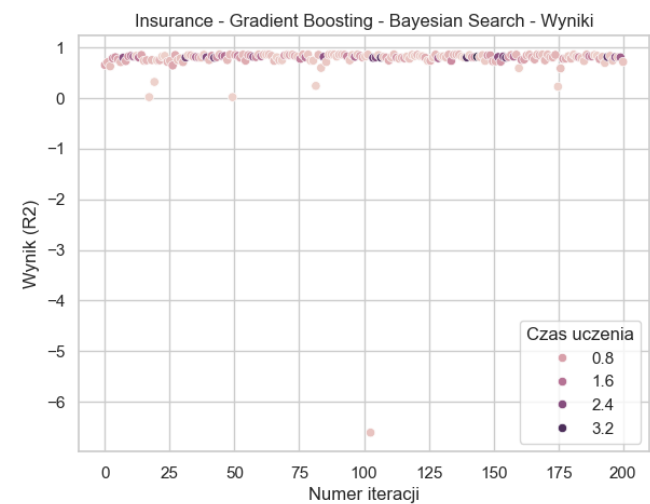
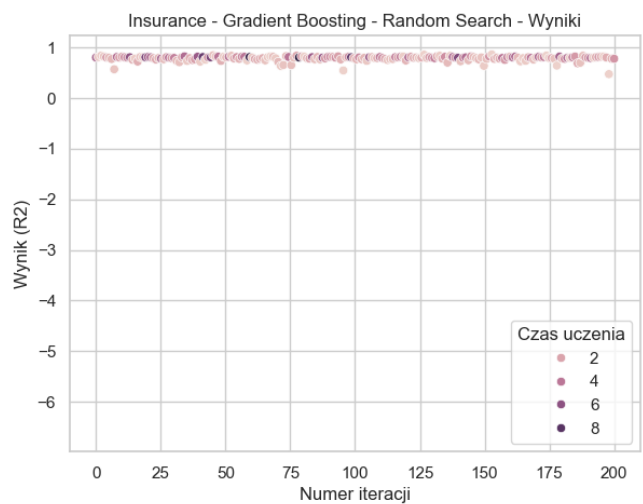
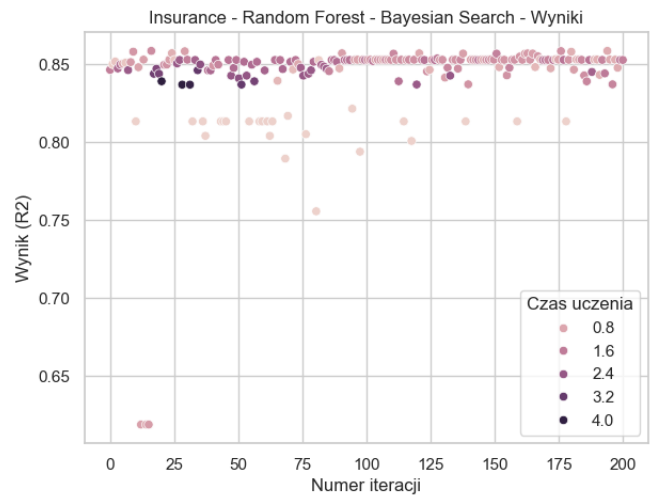
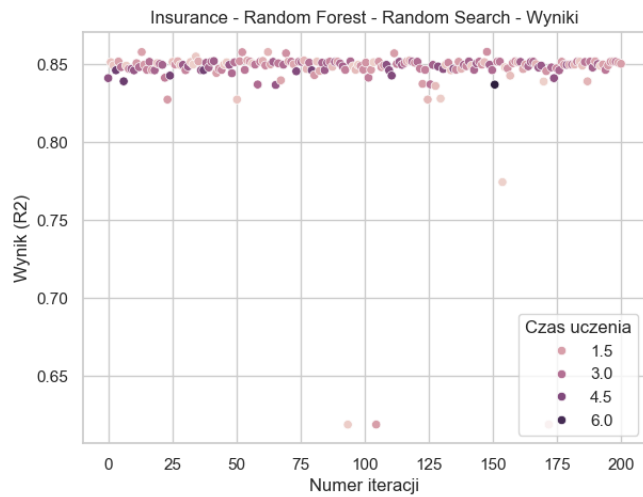
Literatura

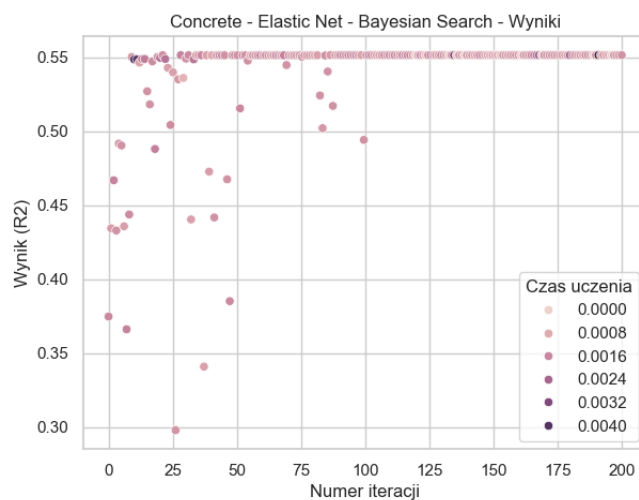
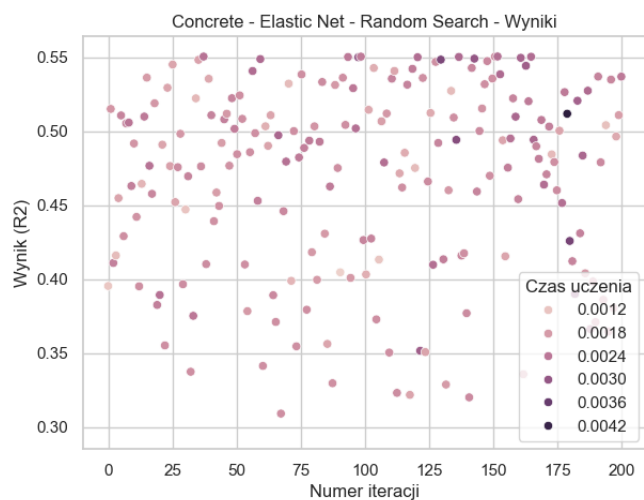
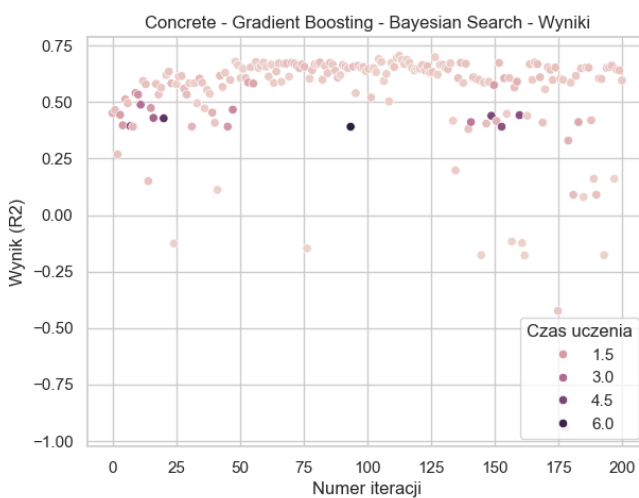
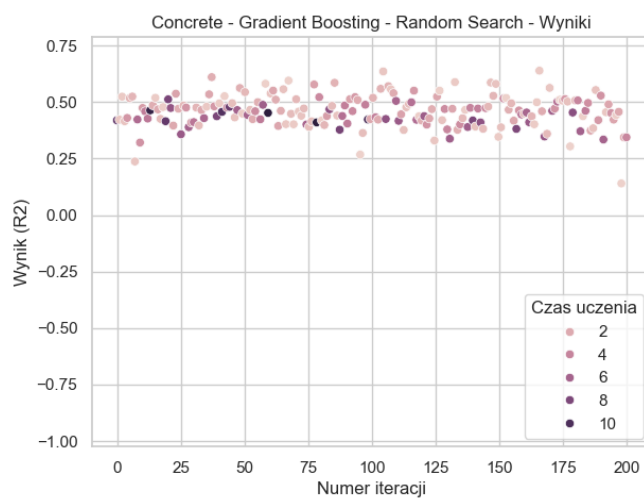
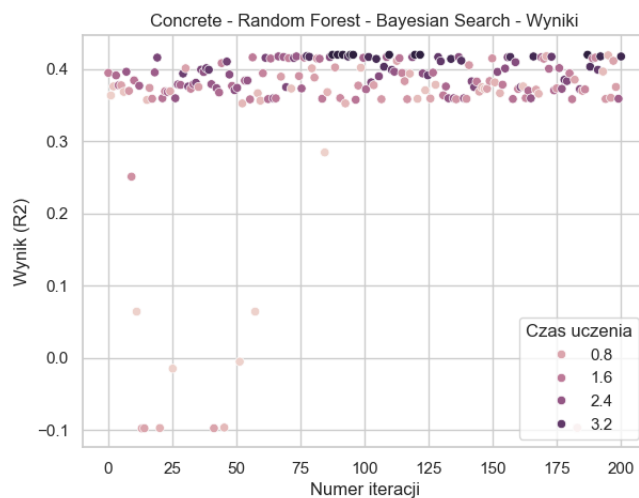
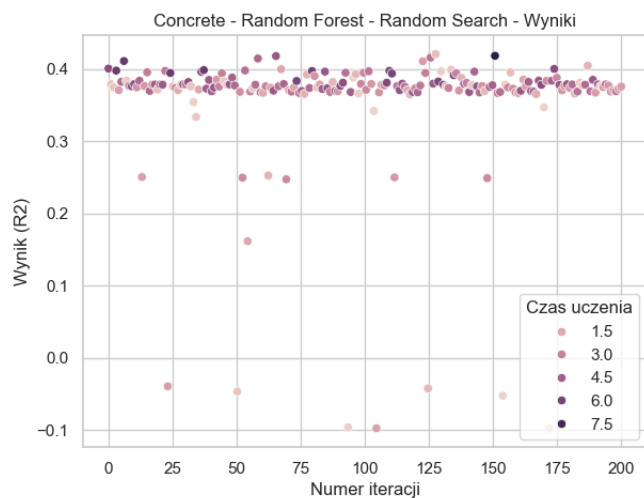
- [1] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *The Journal of Machine Learning Research*, 20(1):1934–1965, 2019.

5 Dodatek

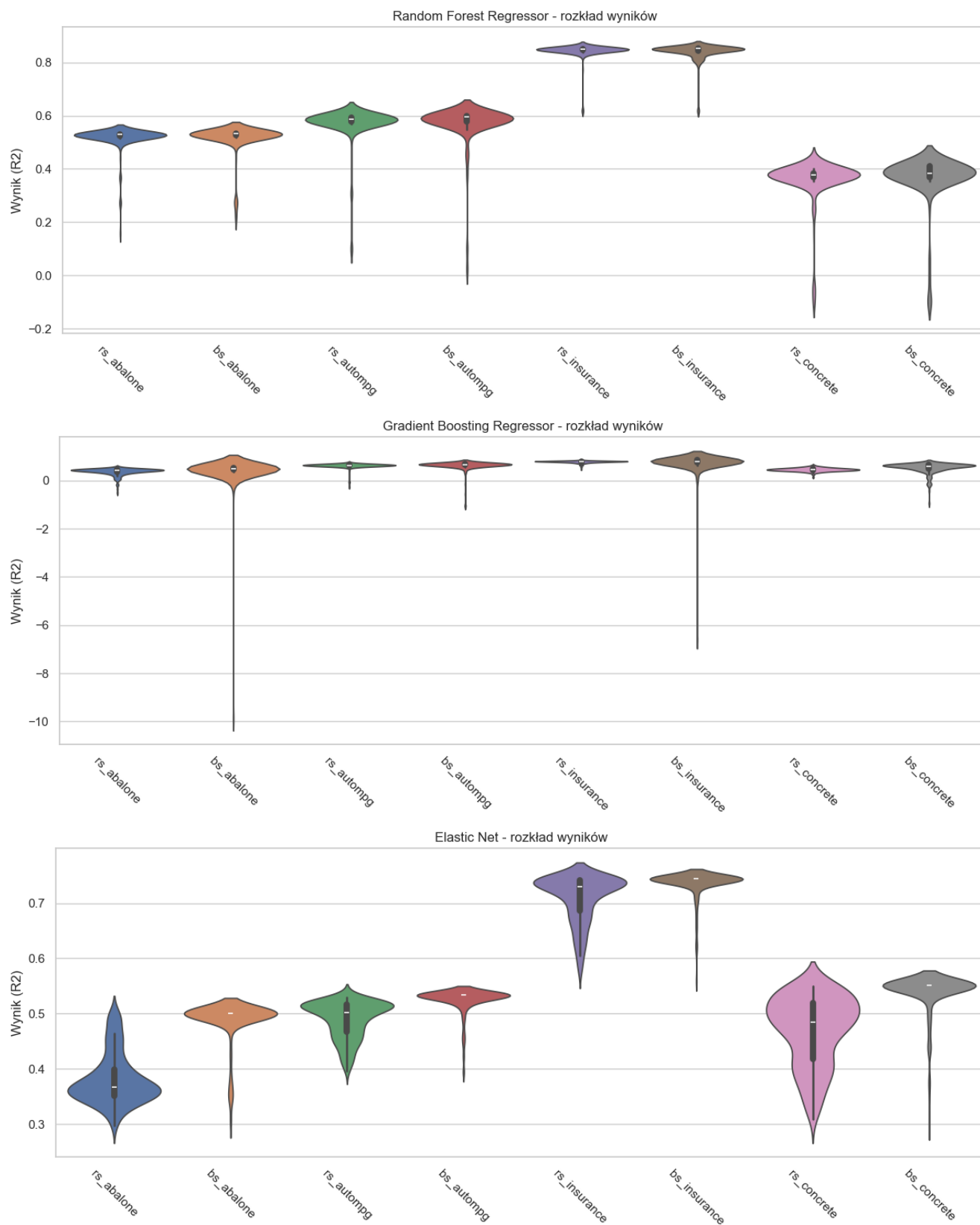




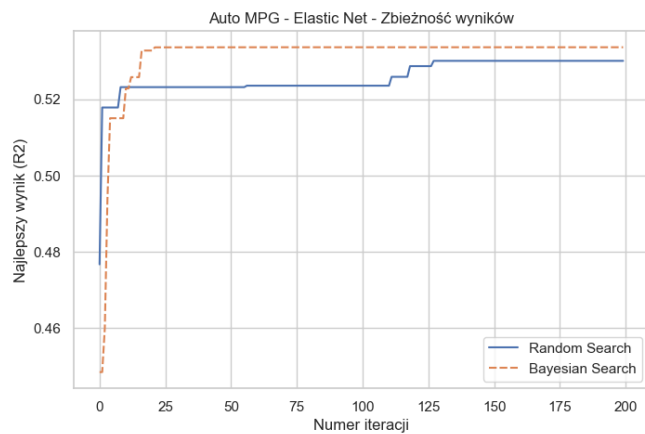
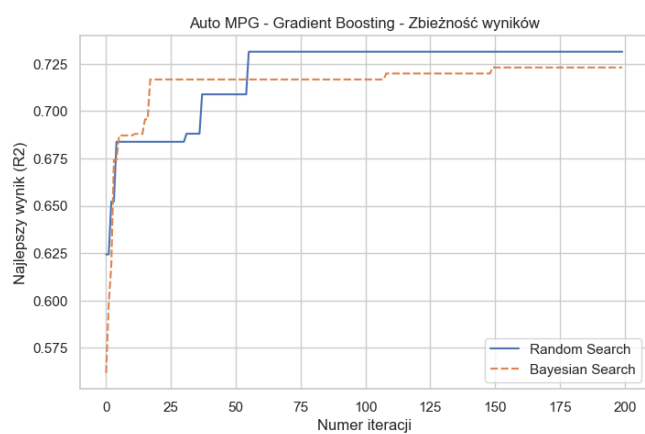
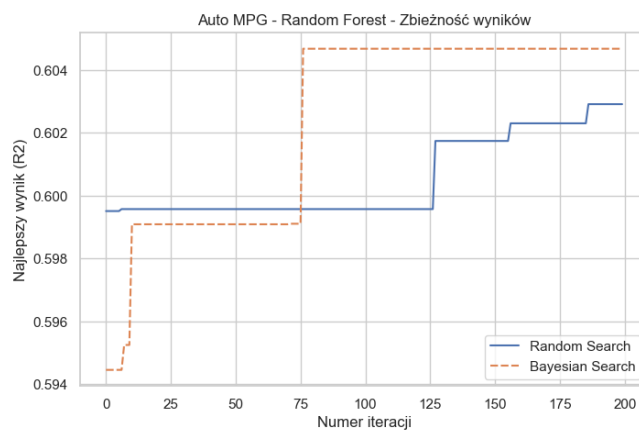
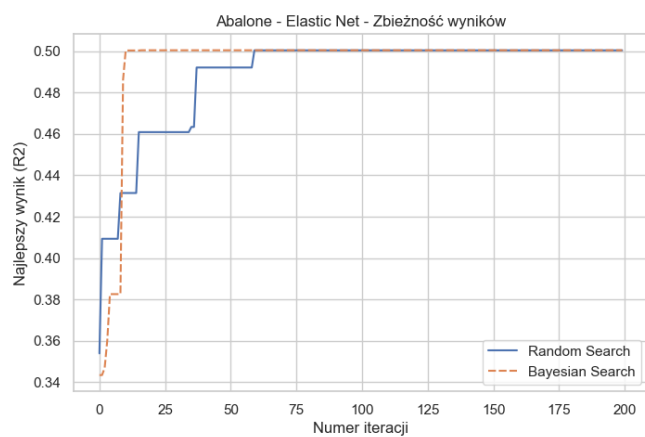
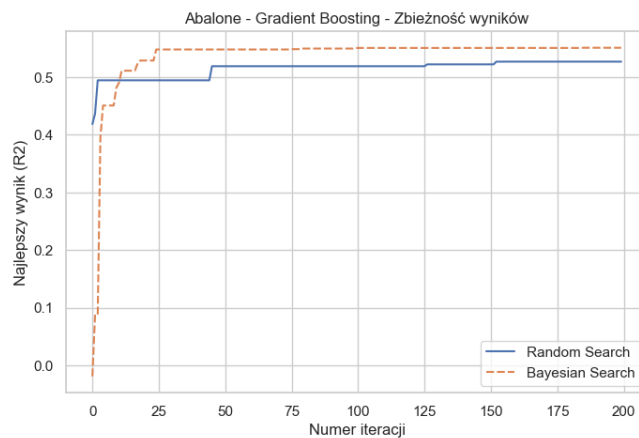
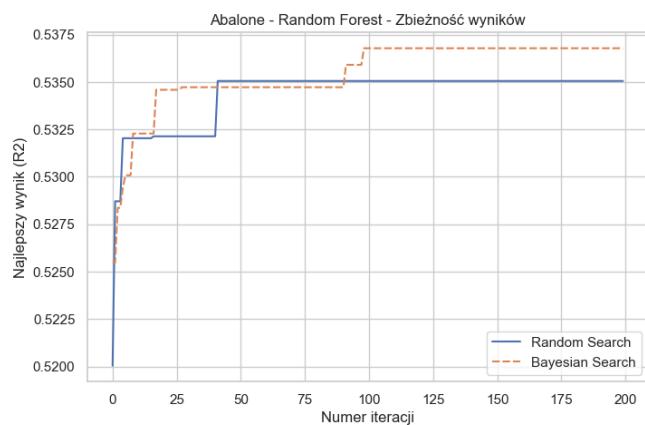


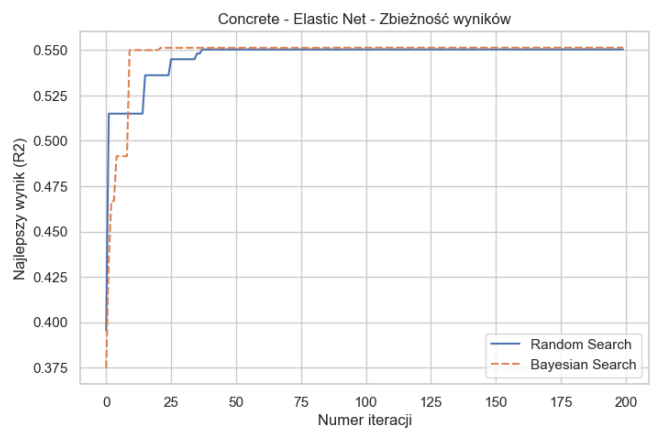
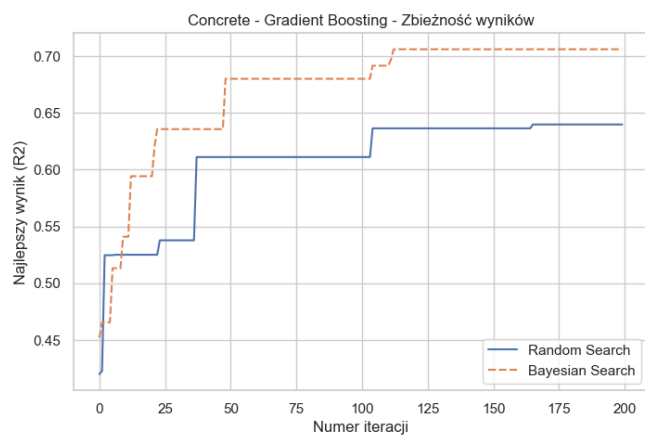
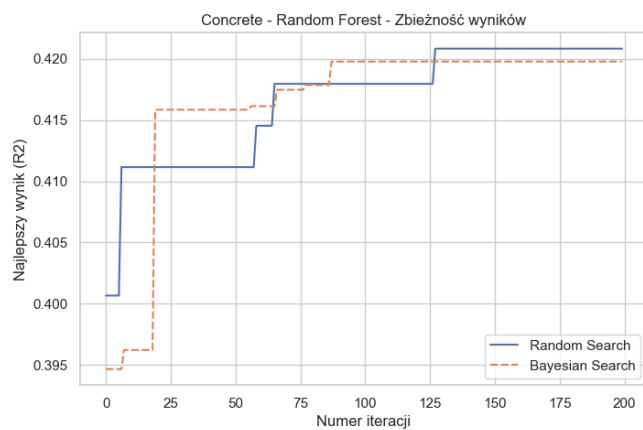
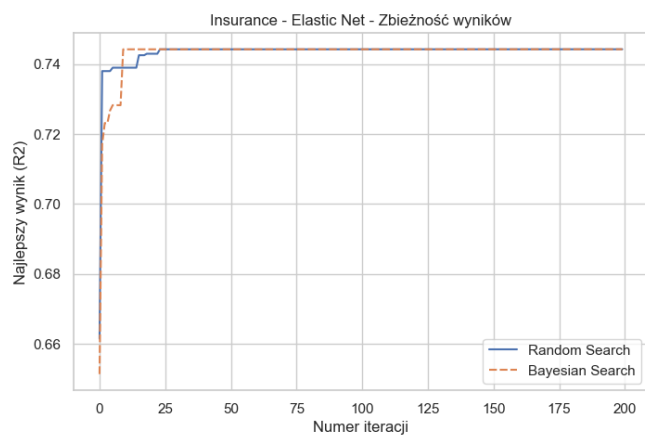
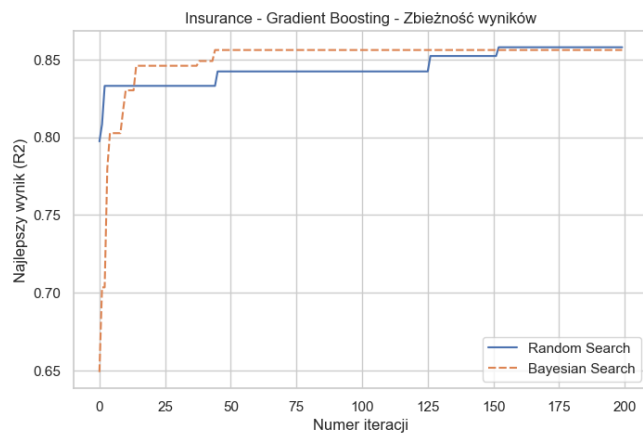
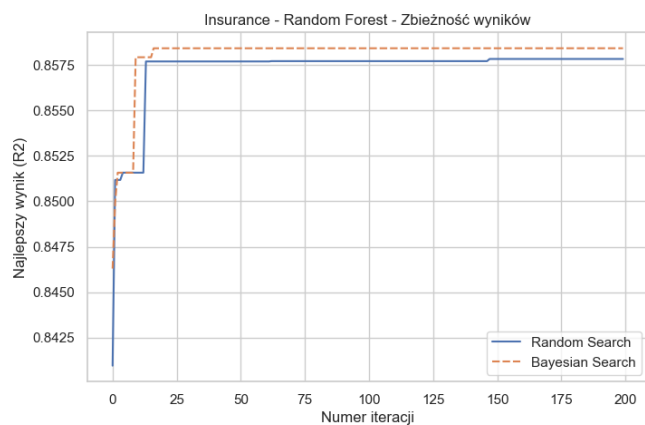


Rysunek 2: Wyniki w poszczególnych iteracjach



Rysunek 3: Rozkłady wyników dla różnych modeli





Rysunek 4: Zbieżność wyników