

AutoML - Projekt 1

333148, 333140, 333295

Ludwik Madej

Karol Kacprzak

Mikołaj Bójski

1 Wprowadzenie

Celem projektu było zbadanie tunowalności modeli regresji logistycznej, lasu losowego i xgboost. W tym celu wybraliśmy 6 zbiorów danych (szczegóły w Appendix), na których przeprowadziliśmy doświadczenia inspirowane artykułem „Tunability: Importance of Hyperparameters of Machine Learning Algorithms”¹, dalej zwanego artykułem.

Dla każdego z modeli przeprowadziliśmy następujący proces:

- W zależności od czasu trenowania algorytmu użyliśmy random search (RS) z dostosowaną liczbą iteracji,
- Analogicznie jak w powyższym punkcie przeprowadziliśmy optymalizację bayesowską (dalej zwaną BO), którą porównaliśmy z RS,
- Wyznaczyliśmy tunowalność modeli oraz zakres optymalnych wartości hiperparametrów,
- Dla dwóch z trzech modeli przeprowadziliśmy procedurę tunowania dla podzbiorów danych o różnych rozmiarach.

2 Tunowanie poszczególnych modeli

Dla każdego z poniższych modeli, wybraliśmy hiperparametry do wyznaczenia wraz ich zakresami inspirując się artykułem. Później używając RS oraz BO zbadaliśmy tunowalność modeli oraz wpływ zmianania wartość hiperparametrów na skuteczność modeli.

¹Philipp Probst, Anne-Laure Boulesteix, Bernd Bischl (2019). *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*. Journal of Machine Learning Research 20:1–32.

2.1 Regresja Logistyczna

Zbadany zakres hiperparametrów jest widoczny z prawej strony wraz z zakresami (\times oznacza, że parametr nie dotyczy modelu z danym penalty). Wszystkie parametry losowane były z rozkładu jednostajnego na zadanym zbiorze (poza C, gdzie $C \sim 2^X$, gdzie $X \sim U[-10, 10]$). W celu określenia jakości modelu wyznaczano średnią wartość roc_auc oraz accuracy za pomocą krosswalidacji używającej StratifiedK-Fold z K=3. Braki danych uzupełniano średnią.

penalty	solver	C	l1_ratio
elasticnet	saga	$[2^{-10}, 2^{10}]$	$[0,1]$
l1	liblinear, saga	$[2^{-10}, 2^{10}]$	\times
l2	lbfgs newton-cg newton-cholesky saga sag liblinear	$[2^{-10}, 2^{10}]$	\times



Później przeprowadzono BO¹ z 4-roma punktami startowymi. Każda BO była przeprowadzana dla ustalonego penalty, metryki $\in (\text{ROC_AUC}, \text{Accuracy})$ oraz impute_strategy='mean'. Po lewej stronie widoczny jest przebieg BO oraz RS - w górnym wierszu (parami) dla ROC_AUC, a w dolnym dla accuracy. Kolejne kolumny odpowiadają hiperparametrowi penalty, w kolejności z powyższej tabeli. Podsumowanie wyników widoczne jest w poniższej tabeli. Zauważamy, że wybierając zestaw hiperparametrów najlepszy dla danego datasetu zyskujemy niewielki przyrost metryki w stosunku do najlepszego globalnie. Dodatkowo używając BO różnica jest niewielka, jednak nie zawsze rozwiązanie BO jest lepsze niż RS.

Dataset	RS						BO	
	ROC_AUC			Accuracy			ROC_AUC	Accuracy
	Global	Dataset	Difference	Global	Dataset	Difference	Global	Global
bank_marketing.csv	0.891637	0.891904	0.000267	0.899892	0.900069	0.000177	0.891021	0.899737
frauds.csv	0.999998	1.000000	0.000002	0.998840	0.999720	0.000880	0.999992	0.999100
loans.csv	0.842799	0.843029	0.000230	0.867404	0.867801	0.000397	0.843038	0.867835
med_visits.csv	0.599597	0.599661	0.000064	0.798067	0.798067	0.000000	0.599637	0.798067
stock_market.csv	0.529317	0.529519	0.000202	0.522643	0.522820	0.000176	0.529521	0.522778

Przeprowadzono również testy statystyczne w celu sprawdzenia, czy rozkłady BO i RS metryk ROC_AUC i Accuracy są takie same, w tym wypadku jedynie dla penalty = elasticnet² z hipotezami: $H_0 : RS = BO$; $H_1 : BO > RS$; $\hat{H}_1 : BO < RS$. W tabeli po prawej widoczne są p-wartości. Widzimy, że wnioski różnią się w zależności od ramki danych i metryki, jednak w większości przypadków odrzucamy hipotezę zerową

na poziomie istotności $\alpha = 0.05$ na korzyść H_1 . Ponadto stwierdzamy, że parametry wybrane globalnie są nieznacznie gorsze od tych wybranych per dataset, co świadczy o niskiej tunowalności algorytmu rzędu 10^{-4} . Podsumowując wyniki jakie model osiągnął poprzez RS wyznaczamy zakresy optymalnych parametrów względem ROC_AUC [Accuracy] : $\log_2 C$: (-0.097404, 9.997078) [(-7.476922, 9.997078)]; $l1_ratio$: (0.018688, 0.993216) [(0.015866, 0.993216)]; penalty : elasticnet [elasticnet, l1]; solver : saga, [saga, liblinear]. Parametry ciągłe zostały wyznaczone poprzez zsumowanie zbiorów 10% najlepszych zestawów hiperparametrów per dataset, a później wyznaczenie $q_{0.05}$ oraz $q_{0.95}$ dla poszczególnych hiperparametrów. W przypadku hiperparametrów kategorycznych wzęto 10% najlepszych hiperparametrów per dataset, a następnie przecięto te zbioru. Zestaw najlepszych hiperparametrów ze względu na roc_auc (accuracy): penalty='l1' ('l1'), solver='liblinear' ('liblinear'), $C=0.11600885868$ (0.0952821484).

¹Optymalizację hiperparametrów przeprowadzono z użyciem skopt (gp_minimize). Model: regresja logistyczna z karą elastic net, oceniana 3-krotną validacją krzyżową na podstawie ROC-AUC. Zastosowano akwizycję EI, $n_{calls} = 100$, $n_{random} = 5$, poziom szumu = 0.15.

²Testy dla pozostałych penalty znajdzie czytelnik w appendixie.

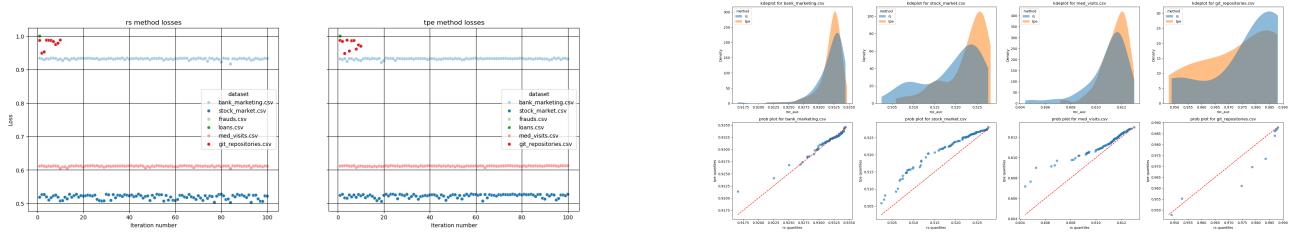
2.2 XGBoost

2.2.1 Wstęp

Przeszukiwana przestrzeń hiperparametrów została zaczerpnięta z tabeli 3 artykułu, ponieważ zakresy proponowane w tabeli 1 okazały się za szerokie. Tunowano wszystkie hiperparametry wypisane dla xgboost w tabeli 3, z wyjątkiem booster, dla którego tylko wartość gbtree pozwalała na określenie wszystkich pozostałych hiperparametrów. Ponadto, dla wybranych zmiennych zastosowano transformację 2^x zgodnie z tabelą 1.

Jako główną miarę ryzyka wybrano ROC AUC, choć część wyników została też uzupełniona o wartość accuracy. Zbiory frauds, loans i git_repositories nie są ujęte wszędzie, gdyż w przypadku pierwszych dwóch dla modelu xgboost uzyskiwano ROC AUC bardzo bliski lub równy 1.0, natomiast ostatni z nich był zbyt wymagający obliczeniowo, dlatego dla niego przeprowadzono 10, zamiast domyślnych 100, iteracji metody optymalizującej. Użyto także StratifiedKFold dla $K = 3$ oraz $early_stopping_rounds = 10$. BO wykonano korzystając z algorytmu TPE z pakietu hyperopt.

Figure 1a i 1b: Na 1a zilustrowano zależność straty (tu: ROC AUC) od numeru iteracji w podziale na rs i tpe. Punkty dla zbiorów frauds, loans i git_repositories są specyficzne z powodów opisanych we Wstępie. Wyniki, szczególnie dla tpe, są bardzo stabilne. Na 1b umieszczone wykresy kde i prob dla ROC AUC w podziale na rs i tpe oraz zbiór danych. Wyniki dla obu metod wydają się dość podobne, szczególnie dla dużych wartości ROC AUC.



2.2.2 Analiza tunowalności

Mediany wyników uzyskanych metodą RS (BO) dla zbiorów danych bank_marketing, med_visits i stock_market wynoszą odpowiednio: 0.9327 (0.9325), 0.6114 (0.6118) i 0.5216 (0.5248). Analiza różnic wyników między RS i BO zostanie szerzej omówiona dalej, w części poświęconej przeprowadzonym testom statystycznym.

Wyznaczono miarę tunowalności, korzystając z miar ryzyka ROC AUC i accuracy dla każdego zbioru osobno, a także zagregowaną (używając średniej lub mediany). Wyniki umieszczone w poniżej tabeli ("mean (median)" dotyczy tylko wiersza "optimal" - dla pojedynczych zbiorów tunowalność niezagregowana).

	colsample	bylevel	colsample_bytree	learning_rate	max_depth	min_child_weight	n_estimators	reg_lambda	reg_alpha	subsample	roc_auc	accuracy	mean (median) roc_auc	tunability	mean (median) accuracy	tunability
optimal	0.640188		0.513647	0.003672	7.000000	1.000000	252.000000	0.028423	1.198657	0.703643	0.690849	0.711741	0.000818 (0.000344)	0.000304 (0.000155)		
optimal lower bound	0.337542		0.428881	0.002017	6.500000	1.150000	956.000000	0.009139	0.002463	0.569834	-	-	-	-	-	
optimal upper bound	0.837110		0.850043	0.053888	13.750000	7.400000	4207.800000	27.842972	4.820166	0.950863	-	-	-	-	-	
bank_marketing	0.480097		0.515544	0.010320	12.000000	1.000000	1896.000000	3.551457	0.078129	0.799071	0.934394	0.907478	0.001781	0.000155		
stock_market	0.591376		0.652274	0.008717	6.000000	2.000000	2647.000000	16.208045	4.922037	0.955299	0.527583	0.520598	0.000330	0.000758		
med_visits	0.494708		0.451425	0.002743	13.000000	2.000000	3020.000000	0.305354	0.262669	0.741330	0.613028	0.798058	0.000344	0.000000		

2.2.3 Testy statystyczne oraz analiza wpływu wielkości zbioru danych

Do porównania różnic wyników, dla miar ryzyka ROC AUC i accuracy, pomiędzy technikami losowania hiperparametrów wykorzystano test Wilcoxona z hipotezami $H_0 : BO = RS$, $H_1 : BO > RS$ przyjmując poziom istotności $\alpha = 0.05$. Testy dla obu miar ryzyka odrzuciły H_0 na korzyść H_1 (p-wartość równa odpowiednio 0.0001 i 0.0004).

Kroki opisane w powyższych sekcjach zostały powtórzone dla podzbiorów danych zawierających 2%, 5% i 10% losowo wybranych obserwacji ze zbiorów. Szczegółowe wyniki i wykresy zostały umieszczone w Appendix.

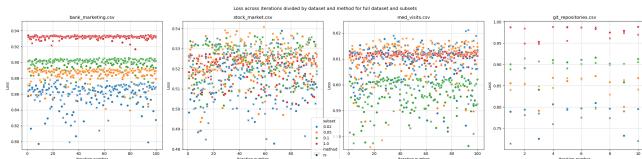


Figure 2: Zależność straty (tu: ROC AUC) od numeru iteracji w podziale na zbiór danych, wielkość rozważanego zbioru (cały lub podzbiory zawierające część obserwacji) oraz rs i tpe. Niewielka liczba punktów dla git_repositories wynika z powodów opisanych we Wstępie. Można zauważyć, że dwa środkowe wykresy są podobne i wyraźnie inne od dwóch skrajnych, co zostało dokładniej opisane w Wynikach.

2.2.4 Wyniki

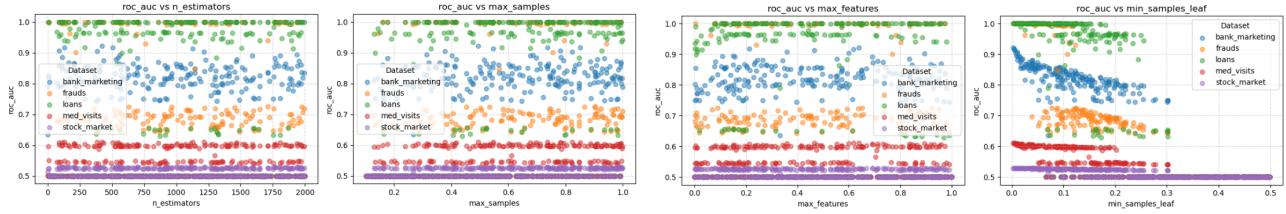
Różnice wyników pomiędzy metodami RS i BO nie są duże, co obrazują wykresy kde na Figure 1b. Jednak test Wilcoxon'a dla ROC AUC i accuracy odrzucił $H_0 : BO = RS$ na korzyść $H_1 : BO > RS$ przy $\alpha = 0.05$. Na Figure 1a można zaobserwować, że stabilne wyniki optymalizacji uzyskujemy po nie więcej niż 20 iteracjach.

Optymalna konfiguracja i zakresy hiperparametrów, umieszczone w tabeli w sekcji Analiza tunowalności, są bliskie wymienionym w artykule, w szczególności pokrywają się zakresy. Algorytm xgboost okazał się niezbyt tunowalny, gdyż miara tunowalności w każdym przypadku jest mniejsza niż 0.002.

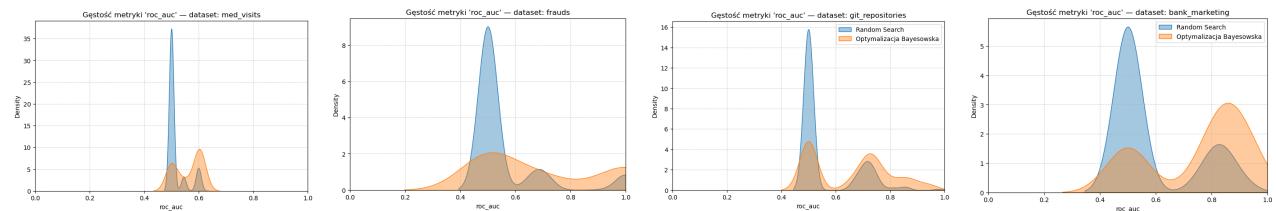
Analiza podzbiorów danych zawierających 2%, 5% i 10% obserwacji wykazała, że dla zbiorów danych med_visits i stock_market przy mniejszej liczbie obserwacji wartość ROC AUC może być nawet wyższa niż dla całego zbioru, jednak dzieje się to kosztem stabilności. Natomiast dla zbiorów bank_marketing i git_repositories większa liczba obserwacji przekłada się na wyższe wyniki, a stabilność jest niewiele mniejsza niż dla pełnych zbiorów. Ponadto, optymalna konfiguracja i zakresy hiperparametrów są wyraźnie inne niż dla całego zbioru i różnią się też między sobą. Dla pozostałych aspektów (wizualna różnica między RS i BO, wyniki testów statystycznych oraz tunowalność) wnioski są zbieżne z tymi dla pełnego zbioru.

2.3 Random Forest

Poszukiwania optymalnego zestawu hiperparametrów przeprowadziliśmy, wzorując się artykułem, na następujących zapisach : n estimators - {1, 2, ..., 2000}, max samples - (0.1, 1), max features - (0, 1), min samples leaf - (0, 1). Ponieważ po wstępnej analizie nie zaobserwowano istotnej różnicy w jakości dla różnych wartości hiperparametru 'criterion', do analizy wykorzystaliśmy jedynie gini. Hiperparametry losowaliśmy z rozkładów jednostajnych , sprawdziliśmy 1000 kombinacji dla 5 zbiorów danych oraz 200 dla 'git_repositories'. Wyznaczyliśmy średnie wartości roc auc i accuracy za pomocą krosswalidacji, używającej StratifiedKFold z K=3.



Następnie przeprowadziliśmy poszukiwania optymalnej kombinacji hiperparametrów dla każdego ze zbiorów danych. Wykonaliśmy 100 iteracji BO (50 dla 'git_repositories') dla metryk roc auc oraz accuracy. porównanie rozkładów otrzymanych metryk dla RS i BO prezentują poniższe wykresy.



Na bazie każdej z metod wyznaczyliśmy optymalne kombinacje hiperparametrów dla każdego zbioru danych. Na wszystkich zbiorach danych najlepsze otrzymane wyniki są zbliżone. Jednak BO nie zawsze znajduje kombinację lepszą od tej wyznaczonej przez RS. Następnie, na podstawie wyników uzyskanych dla RS, wyznaczyliśmy globalnie najlepszą kombinację hiperparametrów, ograniczając się do 200 kombinacji, dla których mieliśmy obliczone wartości metryk dla wszystkich sześciu zbiorów danych. Najlepszą globalnie wyznaczoną kombinacją wartości hiperparametrów dla roc auc (accuracy) okazało się :

n_estimators = 422, max_samples = 0.494191, max_features = 0.639014, min_samples_leaf = 0.002309.

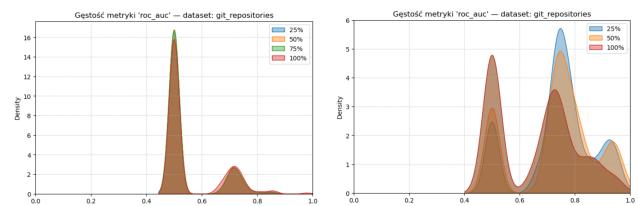
(n_estimators = 923, max_samples = 0.868617, max_features = 0.588207, min_samples_leaf = 0.000186)

Algorytm okazał się słabo tunowalny, w każdym przypadku poprawa wyniku względem wyznaczonych globalnie optymalnych wartości okazała się niewielka. Jedynym hiperparametrem nie mieszczącym się w zakresie sugerowanym w artykule jest min_samples_leaf osiągając wartość znacznie poniżej dolnej granicy zaproponowanej w artykule.

dataset	Global					RS					BO				
	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC				
bank_marketing	0.934516	923	0.868617	0.588207	0.000186	0.934516	1812	0.530724	0.468840	0.000141	0.935052				
frauds	0.999998	908	0.834912	0.105278	0.066559	1.000000	1866	0.354271	0.993508	0.033331	1.000000				
git_repositories	0.978497	923	0.868617	0.588207	0.000186	0.978497	573	0.380384	0.546081	0.000471	0.947219				
loans	1.000000	820	0.959409	0.724893	0.034337	1.000000	980	0.964899	0.433191	0.006062	1.000000				
med_visits	0.610929	422	0.494191	0.639014	0.002309	0.610929	1897	0.685562	0.326805	0.003156	0.610898				
stock_market	0.521478	1339	0.214921	0.257563	0.003181	0.529788	609	0.642639	0.133615	0.008649	0.529222				
	Accuracy	n_estimators	max_samples	max_features	min_samples_leaf	Accuracy	n_estimators	max_samples	max_features	min_samples_leaf	Accuracy				
bank_marketing	0.907899	923	0.868617	0.588207	0.000186	0.907899	1580	0.576887	0.594955	0.000362	0.907191				
frauds	0.999100	547	0.697259	0.836245	0.018161	1.000000	1656	0.751759	0.634408	0.002352	0.999000				
git_repositories	0.964803	923	0.868617	0.588207	0.000186	0.964803	146	0.666140	0.357867	0.000198	0.960098				
loans	0.999993	90	0.780468	0.359204	0.010935	1.000000	460	0.354123	0.327526	0.000532	1.000000				
med_visits	0.798067	179	0.494991	0.858598	0.697368	0.798067	337	0.261448	0.759370	0.960183	0.798067				
stock_market	0.515615	507	0.851613	0.213075	0.004306	0.521335	2000	0.799498	0.366833	0.002781	0.521086				

Wykonaliśmy test statystyczny Wilcoxona z hipotezami $H_0 : RS = BO, H_1 : RS < BO$ na poziomie istotności $\alpha = 0.05$ z podziałem na zbiorów danych dla accuracy oraz roc auc. Dla 5 z 6 zbiorów danych efektem było odrzucenie hipotezy zerowej dla accuracy i dla każdego zbioru dla roc auc.

Następnie powtórzyliśmy procedurę dla 25%, 50% i 75% (tylko RS) obserwacji ze zbiorów danych. Dla RS (sto kombinacji hiperparametrów) rozkład otrzymany wyników średnich ROC AUC oraz Accuracy nie zmienił się istotnie, natomiast dla BO dla niektórych zbiorów danych wyższe wartości były osiągane częściej (Appendix).



3 Appendix

3.1 Datasetsy

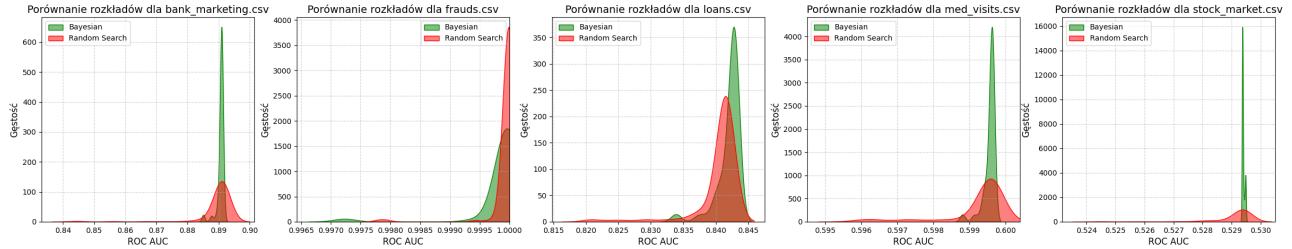
- stock_market - numerai28.6: <https://www.openml.org/search?type=data&status=active&id=23517>
- bank_marketing - bank-marketing: <https://www.openml.org/search?type=data&status=active&id=1461>
- loans - <https://www.kaggle.com/datasets/yassersh/loan-default-dataset>
- frauds - <https://www.kaggle.com/datasets/samayashar/fraud-detection-transactions-dataset>
- git_repositories - <https://www.openml.org/search?type=data&status=active&id=43357&sort=runs>
- med_visits - <https://www.openml.org/search?type=data&status=active&id=43439>

3.2 Regresja Logistyczna

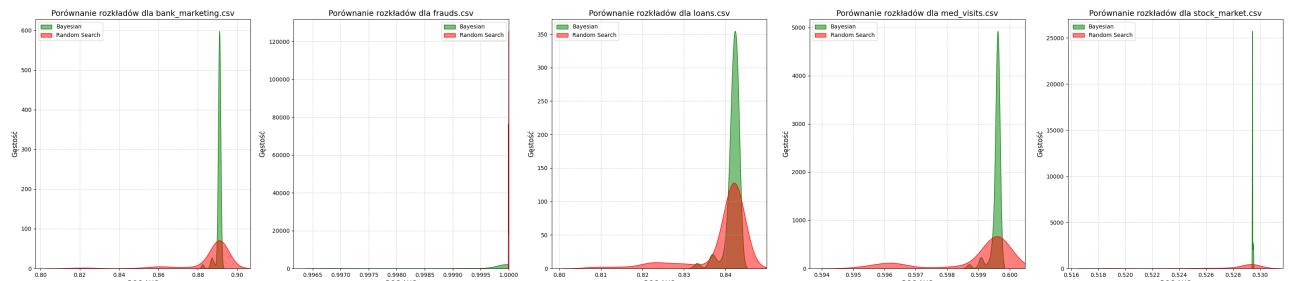
Przyjrzyjmy się wspomnianym wcześniej wynikom testów statystycznych, których użyliśmy do sprawdzenia: $H_0 : RS = BO$; $H_1 : BO > RS$; $\hat{H}_1 : BO < RS$. Poniżej tabela z wynikami testu wilcoxona dla ostatnich 70 iteracji optymalizacji bayesowskiej oraz pierwszych 70 iteracji random search.

Dataset	ElasticNet				L1				L2			
	ROC_AUC		Accuracy		ROC_AUC		Accuracy		ROC_AUC		Accuracy	
	H_1	\hat{H}_1										
bank_marketing	$9.999e^{-1}$	$4.362e^{-5}$	$9.944e^{-1}$	$5.615e^{-3}$	$4.022e^{-1}$	$5.978e^{-1}$	$9.953e^{-1}$	$4.725e^{-3}$	$8.395e^{-1}$	$1.605e^{-1}$	$7.448e^{-1}$	$2.552e^{-1}$
frauds	$2.300e^{-5}$	$9.999e^{-1}$	$5.761e^{-1}$	$4.239e^{-1}$	1	0	$1.365e^{-1}$	$8.635e^{-1}$	$5.678e^{-1}$	$4.322e^{-1}$	$3.450e^{-1}$	$6.550e^{-1}$
loans	0	1	$2.452e^{-2}$	$9.755e^{-1}$	$5.919e^{-3}$	$9.941e^{-1}$	$7.179e^{-4}$	$9.993e^{-1}$	$6.359e^{-1}$	$3.641e^{-1}$	$5.185e^{-1}$	$4.815e^{-1}$
med_visits	$1.000e^{-8}$	$9.999e^{-1}$	1	1	$1.424e^{-5}$	$9.999e^{-1}$	1	1	$5.255e^{-1}$	$4.745e^{-1}$	1	1
stock_market	0	1	$6.469e^{-1}$	$3.531e^{-1}$	0	1	$1.645e^{-1}$	$8.355e^{-1}$	$8.451e^{-1}$	$1.549e^{-1}$	$7.612e^{-1}$	$2.388e^{-1}$

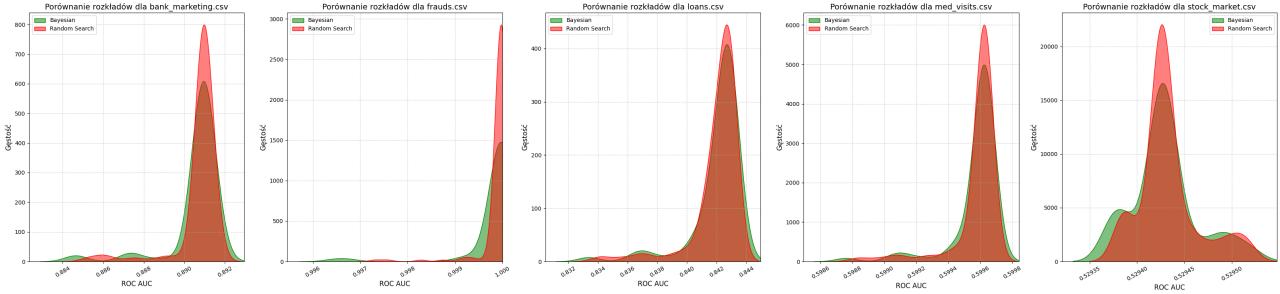
Teraz zwizualizujmy sobie rozkłady metryk roc_auc oraz accuracy dla optymalizacji bayesowskiej oraz random search. Podobnie jak we wcześniejszych rozważaniach rozbijamy problem ze względu na penalty.



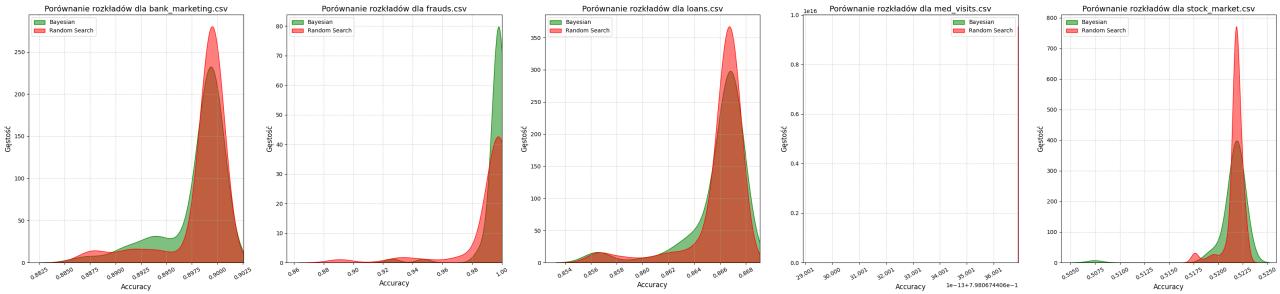
Gęstości metryk roc_auc dla penalty = 'elasticnet'



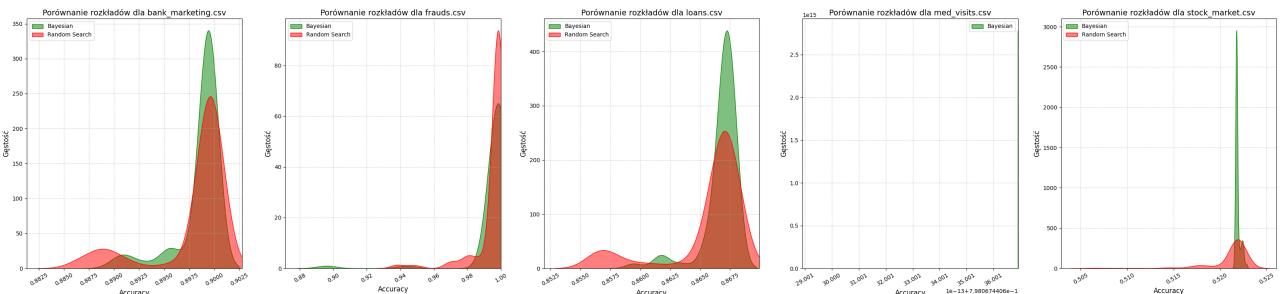
Gęstości metryk roc_auc dla penalty = 'l1'



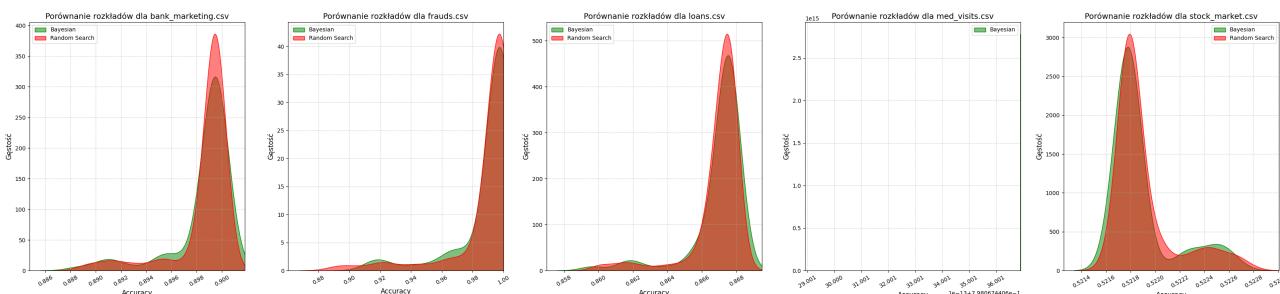
Gęstości metryk roc_auc dla penalty = 'l2'



Gęstości metryk accuracy dla penalty = 'elasticnet'

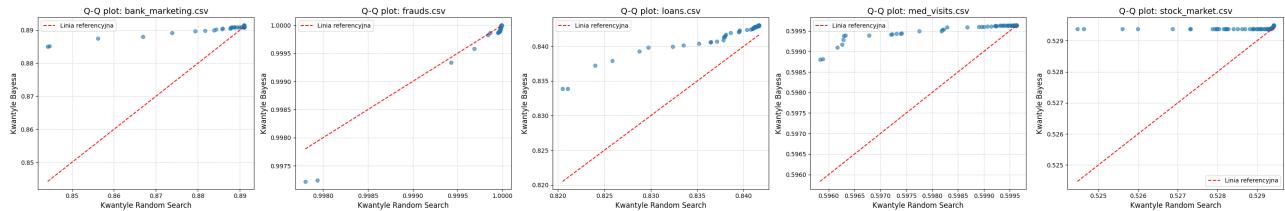


Gęstości metryk accuracy dla penalty = 'l1'

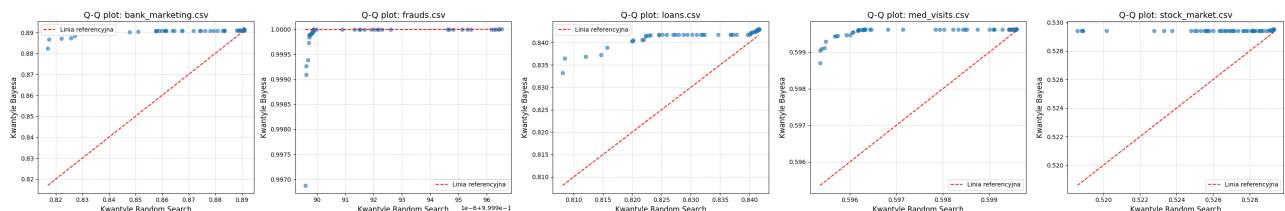


Gęstości metryk accuracy dla penalty = 'l2'

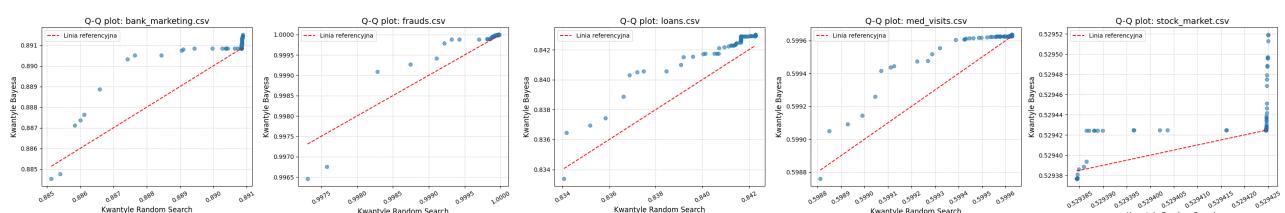
Widzimy, że rozkłady są bardzo podobne. Puste wykresy dla accuracy spowodowane są stałym rozkładem obydwu metryk. Teraz porównajmy wykresy Q-Q dla random search oraz optymalizacji bayesowskiej.



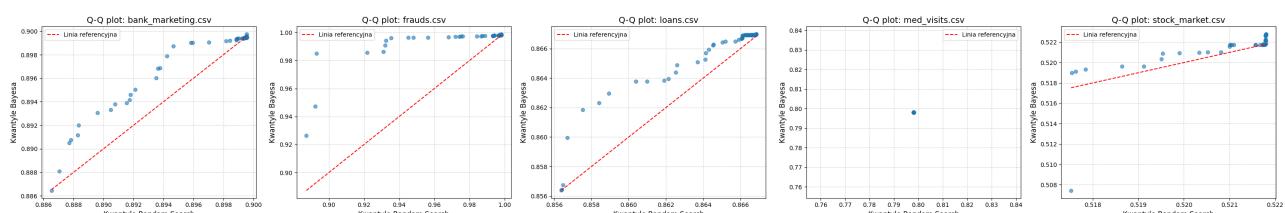
Wykresy Q–Q dla metryki ROC AUC, penalty = 'elasticnet'



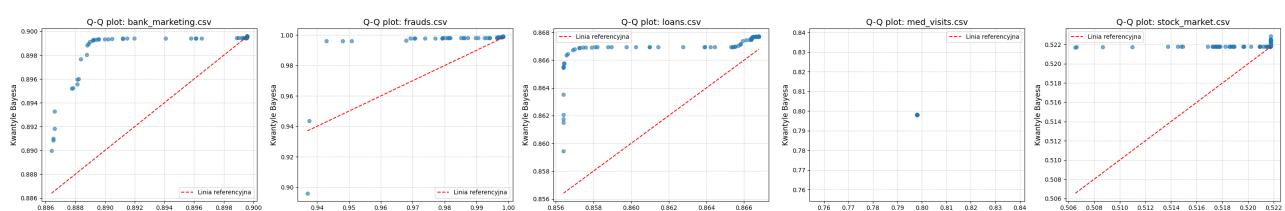
Wykresy Q–Q dla metryki ROC AUC, penalty = 'l1'



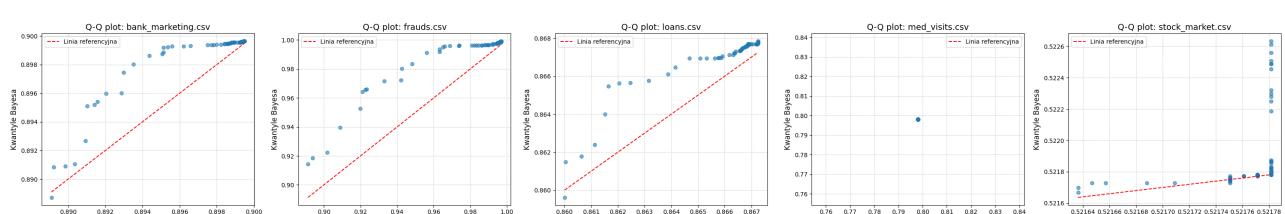
Wykresy Q–Q dla metryki ROC AUC, penalty = 'l2'



Wykresy Q–Q dla metryki accuracy, penalty = 'elasticnet'



Wykresy Q–Q dla metryki accuracy, penalty = 'l1'



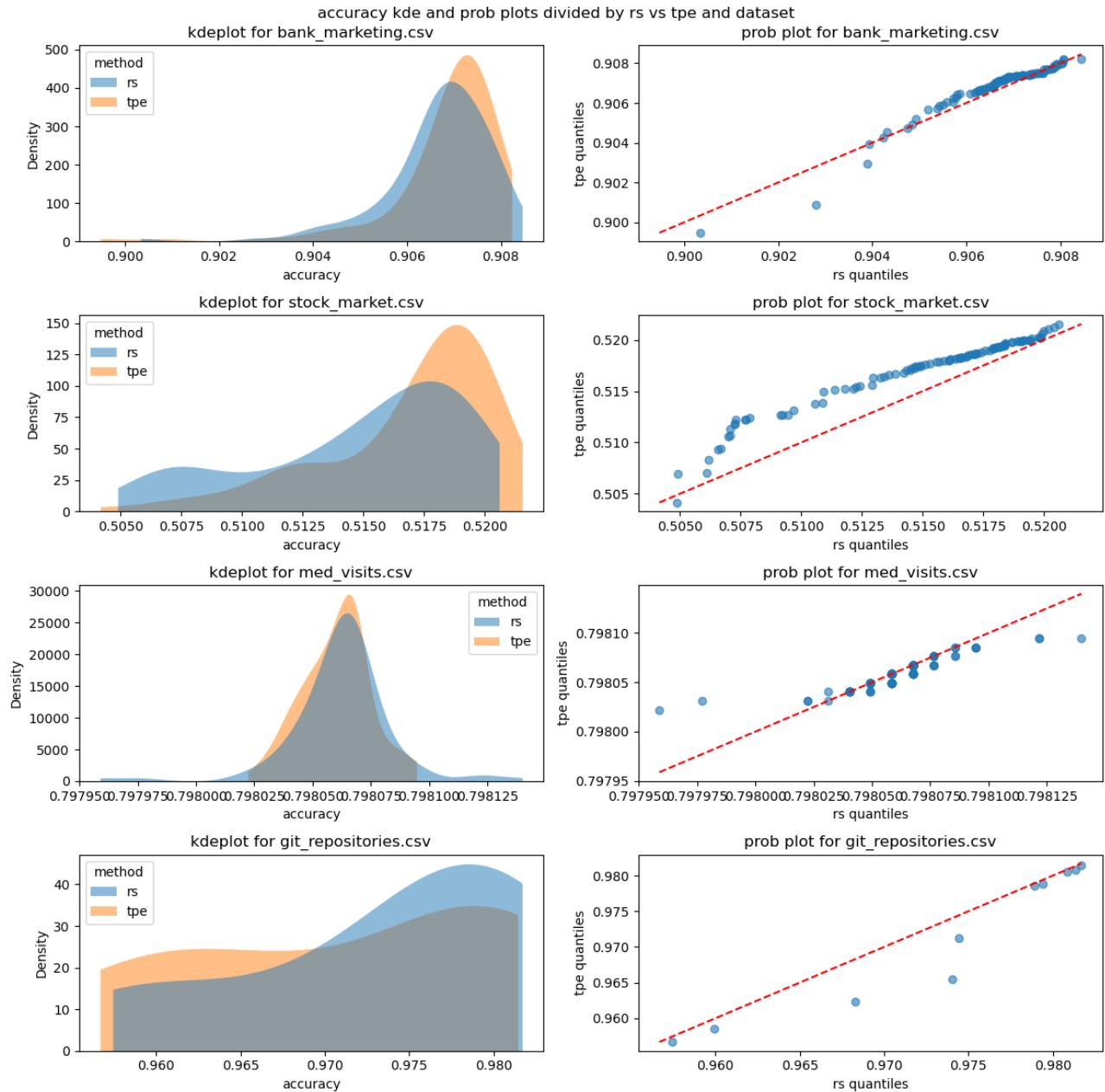
Wykresy Q–Q dla metryki accuracy, penalty = 'l2'

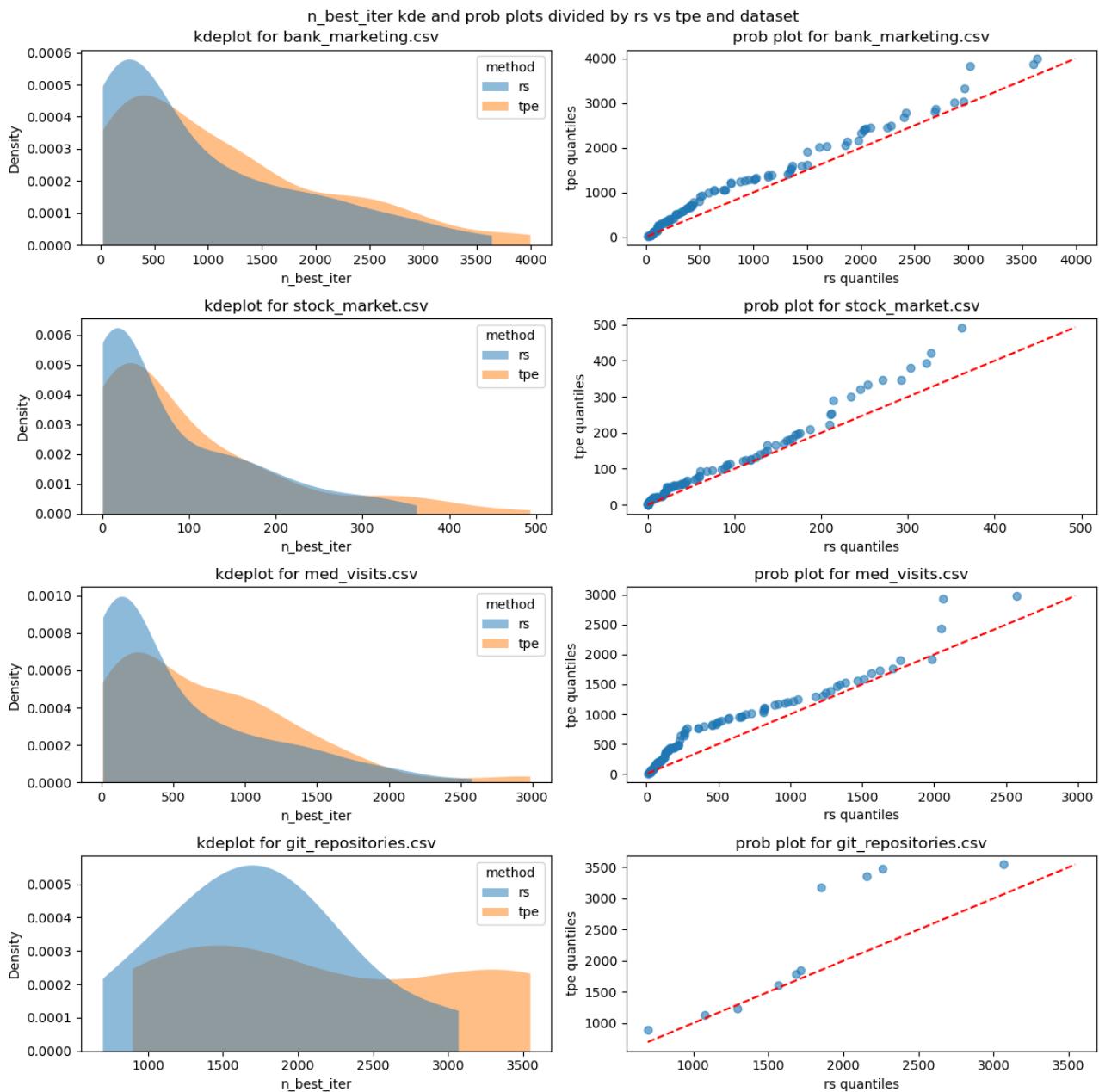
Zauważamy, że optymalizacja bayesowska na tych wykresach wydaje się znacznie lepsza od losowego przeszukiwania przestrzeni hiperparametrów. Nawet w przypadku zbioru bank_marketing, gdzie RS hipotetycznie był lepszy widoczna jest dominacja metody bayesowskiej.

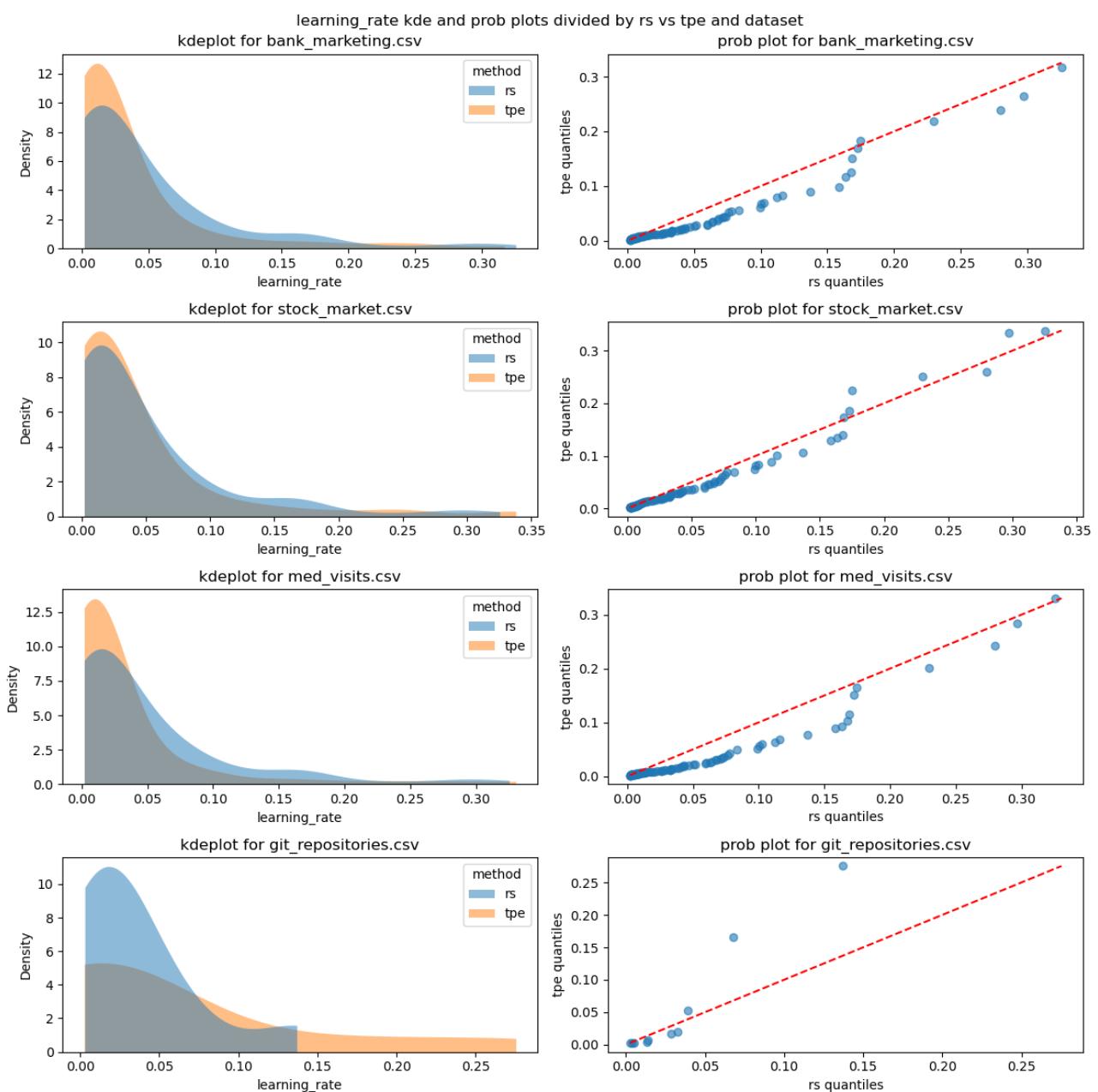
3.3 XGBoost

3.3.1 Pozostałe wykresy dla pełnego zbioru danych

Poniższe trzy wykresy są odpowiednikami Figure 1b dla accuracy, n_best_iter i learning_rate.







3.3.2 Analiza tunowalności dla podzbiorów danych zawierających 10%, 5% i 2% obserwacji

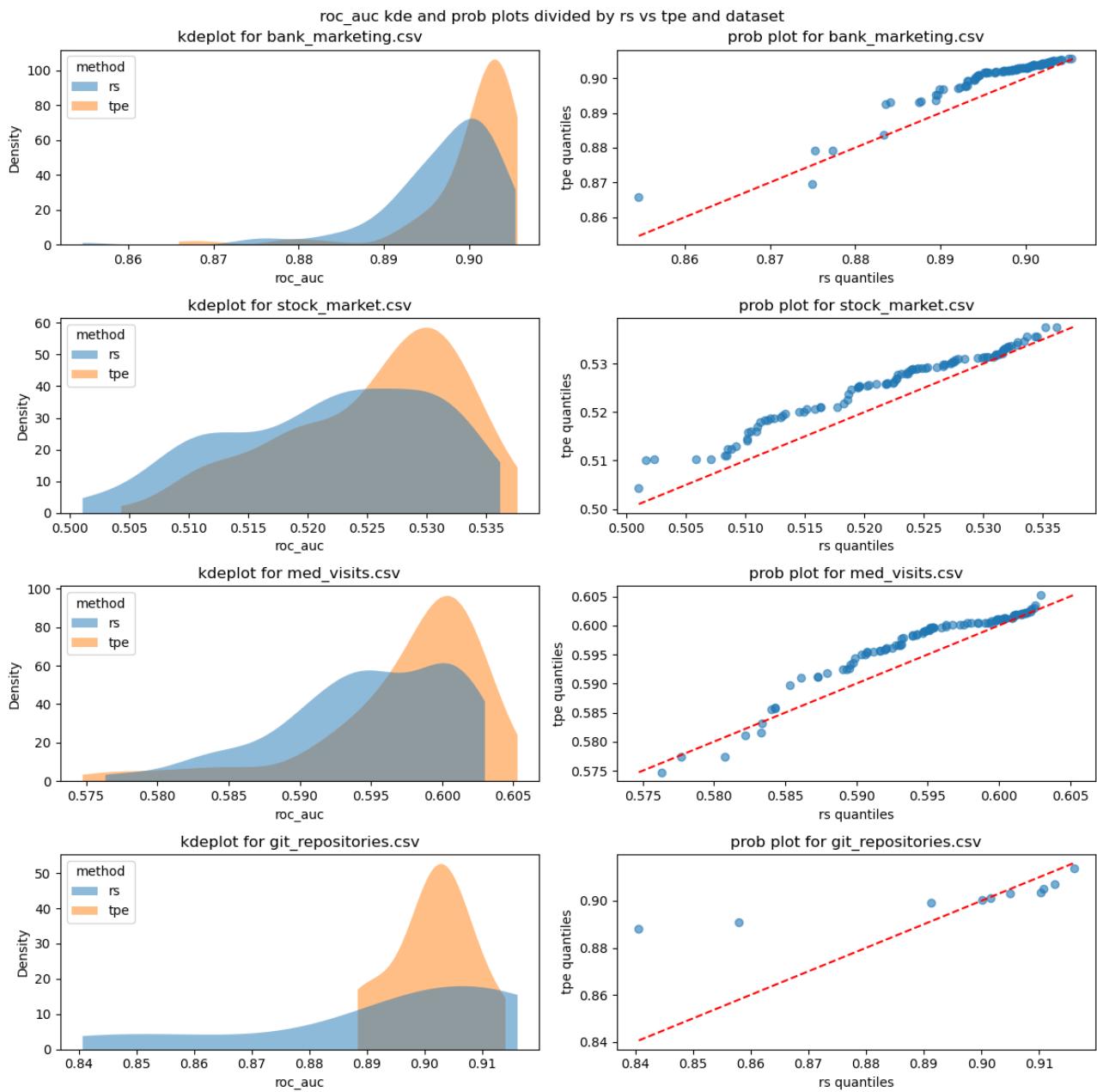
	colsample	bylevel	colsample	bytree	learning_rate	max_depth	min_child_weight	weight	n_estimators	reg_lambda	reg_alpha	subsample	roc_auc	accuracy	mean (median) roc_auc_tunability	mean (median) accuracy_tunability
optimal	0.621188	-	0.5153847	-	0.002092	-	5.000000	-	2522.000000	0.0228423	0.002097	0.740843	0.677416	0.735589	0.004089 (0.01183)	0.006516 (0.0)
optimal lower_bound	0.529811	0.425186	0.002069	6.400000	1.150000	1030.250000	0.000651	0.002064	0.562814	-	-	-	-	-	-	-
optimal upper_bound	0.857987	0.843060	0.232283	13.600000	7.400000	4408.700000	24.810167	3.440541	0.888757	-	-	-	-	-	-	-
bank_marketing	0.510281	0.433260	0.002867	6.000000	2.000000	2693.000000	0.287081	0.037071	0.678663	0.905410	0.892723	0.004183	-	-	0.001548	-
stock_market	0.640188	0.513647	0.003672	7.000000	4.000000	2522.000000	0.028423	1.198657	0.703643	0.536136	0.517546	-	-	-	0.000000	-
med_visits	0.468368	0.854498	0.026394	8.000000	2.000000	3121.000000	4.938188	0.122709	0.631604	0.602967	0.798046	0.008083	-	-	0.000000	-
	colsample	bylevel	colsample	bytree	learning_rate	max_depth	min_child_weight	weight	n_estimators	reg_lambda	reg_alpha	subsample	roc_auc	accuracy	mean (median) roc_auc_tunability	mean (median) accuracy_tunability
optimal	0.456855	0.816370	0.002203	11.000000	4.000000	2767.000000	3.295367	2.609028	0.925101	0.679803	0.736948	0.002877 (0.003360)	-	-	-0.000829 (0.0)	-
optimal lower_bound	0.341843	0.504841	0.002128	6.300000	1.150000	993.950000	0.136298	0.002784	0.552078	-	-	-	-	-	-	-
optimal upper_bound	0.852270	0.861877	0.100324	13.700000	7.400000	4418.500000	27.857057	5.292225	0.937879	-	-	-	-	-	-	-
bank_marketing	0.474399	0.803700	0.014171	14.000000	2.000000	1294.000000	0.033106	1.700614	0.680060	0.592386	0.892036	0.003360	-	-	0.000003	-
stock_market	0.852499	0.692474	0.060226	11.000000	2.000000	3733.000000	27.295885	0.464419	0.739679	0.538649	0.518274	0.004158	-	-	-0.002490	-
med_visits	0.471079	0.777212	0.015050	13.000000	4.000000	1023.000000	6.685887	0.009805	0.622343	0.617003	0.798046	0.001112	-	-	0.000000	-
	colsample	bylevel	colsample	bytree	learning_rate	max_depth	min_child_weight	weight	n_estimators	reg_lambda	reg_alpha	subsample	roc_auc	accuracy	mean (median) roc_auc_tunability	mean (median) accuracy_tunability
optimal	0.409011	0.631619	0.168704	13.000000	2.000000	1053.000000	3.530435	0.016344	0.507452	0.669332	0.736507	0.007262 (0.006673)	-	-	0.001041 (0.000007)	-
optimal lower_bound	0.383237	0.428537	0.002478	6.400000	1.150000	1134.000000	0.009822	0.002808	0.556012	-	-	-	-	-	-	-
optimal upper_bound	0.823722	0.850810	0.232530	13.600000	7.400000	4301.800000	20.800853	4.003574	0.957607	-	-	-	-	-	-	-
bank_marketing	0.545012	0.457651	0.008554	9.000000	2.000000	2415.000000	0.010885	0.234408	0.766580	0.873264	0.891609	0.011056	-	-	0.000007	-
stock_market	0.718274	0.734710	0.011837	10.000000	2.000000	2421.000000	7.203754	0.099044	0.701839	0.538854	0.522845	0.006673	-	-	0.003115	-
med_visits	0.458121	0.688479	0.302288	10.000000	2.000000	4054.000000	1.356009	0.075406	0.668609	0.671663	0.798190	0.004057	-	-	0.000000	-

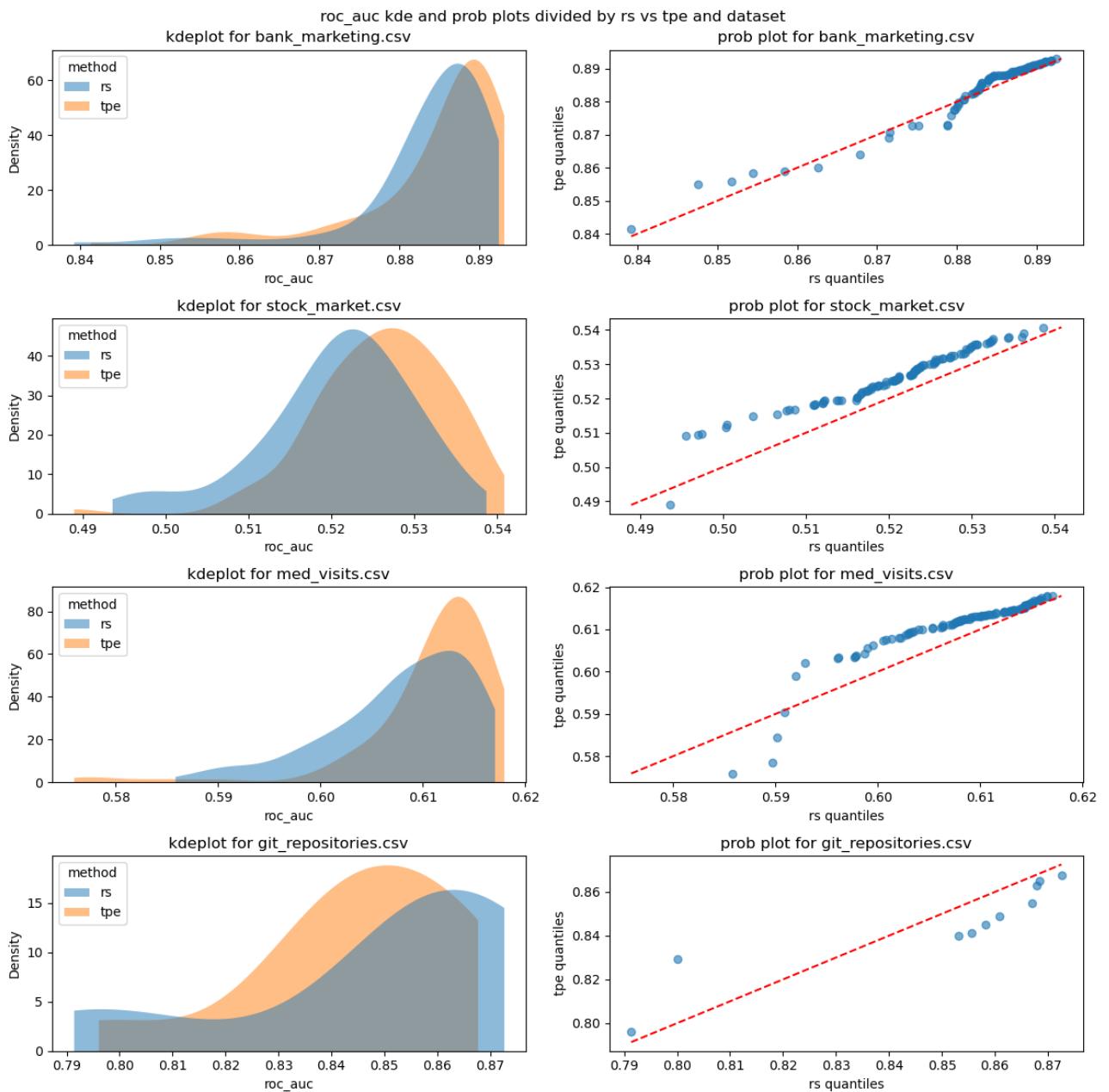
3.3.3 Testy statystyczne dla podzbiorów danych zawierających 10%, 5% i 2% obserwacji

Wykorzystano test Wilcooxona z identycznymi hipotezami i poziomem istotności jak dla całego zbioru danych. We wszystkich przypadkach, zarówno dla ROC AUC, jak i accuracy, test odrzucił $H_0 : BO = RS$ na korzyść $H_1 : BO > RS$ przy $\alpha = 0.05$. Dla podzbiorów danych zawierających 10%, 5% i 2% obserwacji p-wartości przy testowaniu ROC AUC (accuracy) wyniosły odpowiednio $1.3e-09$ (0.0018), $6.8e-09$ (0.0036) i 0.00013 (0.0446).

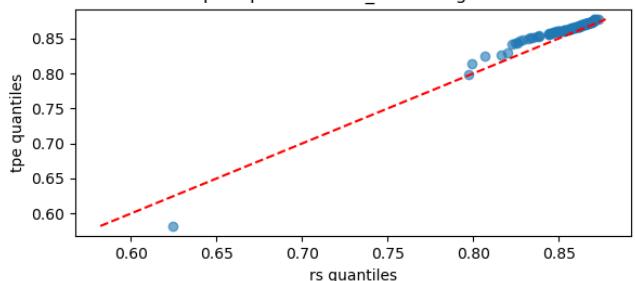
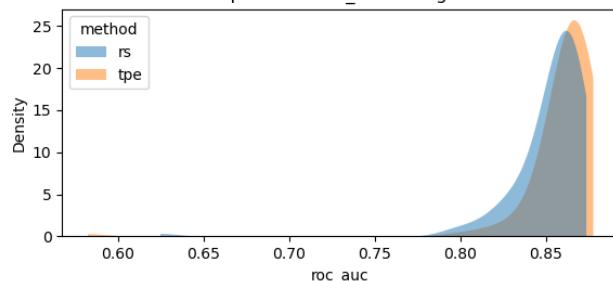
3.3.4 Wykresy dla podzbiorów danych

Poniższe trzy wykresy są odpowiednikami Figure 1b dla podzbiorów danych zawierających odpowiednio 10%, 5% i 2% obserwacji.

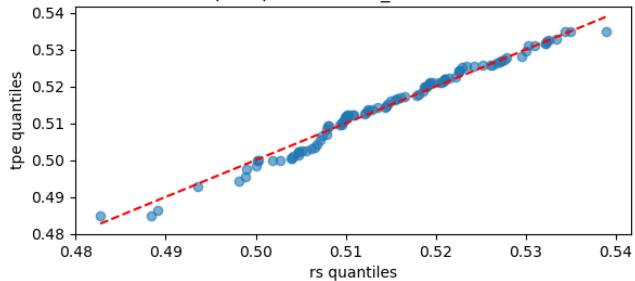
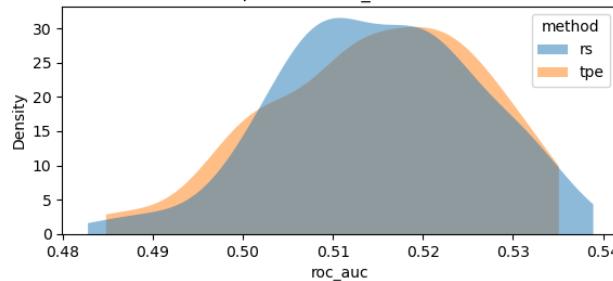




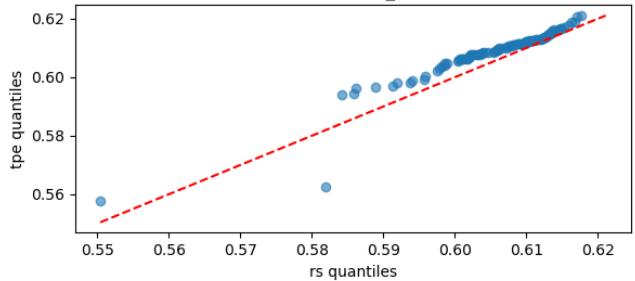
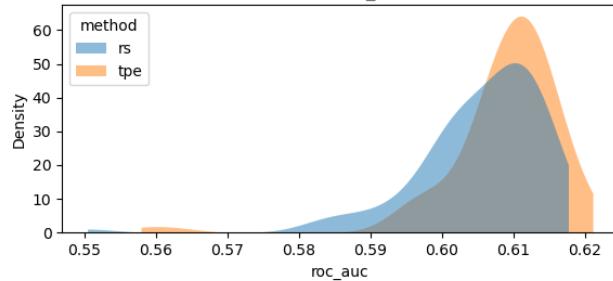
roc_auc kde and prob plots divided by rs vs tpe and dataset
kdeplot for bank_marketing.csv



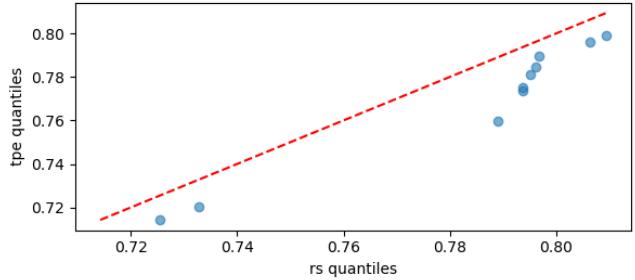
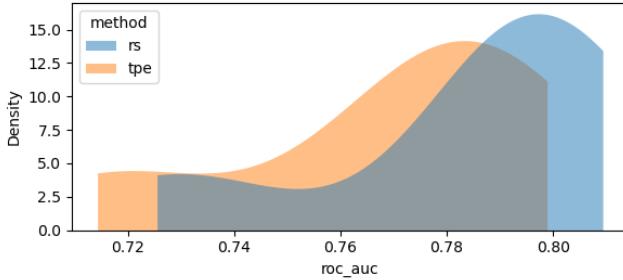
kdeplot for stock_market.csv



kdeplot for med_visits.csv



kdeplot for git_repositories.csv



3.4 Random Forest

3.4.1 Stabilność wyników dla podzbiorów

dataset	Global						RS						BO					
	ROC AUC						Accuracy											
	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC	Accuracy	n_estimators	max_samples	max_features	min_samples_leaf	Accuracy	
bank_marketing	0.908046		892	0.429435	0.523481	0.006785	0.908046		1025	0.901026	0.230610	0.000360	0.926164					
frauds	1.000000		892	0.429435	0.523481	0.006785	1.000000		1198	0.948098	0.469106	0.067935	1.000000					
git_repositories	0.847440		892	0.429435	0.523481	0.006785	0.847440		1709	0.660053	0.726557	0.000121	0.964627					
loans	1.000000		1098	0.720124	0.555097	0.042012	1.000000		1971	0.510516	0.922871	0.007884	1.000000					
med_visits	0.606776		892	0.429435	0.523481	0.006785	0.606776		140	0.343746	0.606457	0.011413	0.607476					
stock_market	0.525248		541	0.413956	0.846941	0.050349	0.525766		639	0.498396	0.117619	0.027225	0.527950					
	Accuracy																	
bank_marketing	0.897894		892	0.429435	0.523481	0.006785	0.897894		1438	0.421139	0.360619	0.000119	0.902584					
frauds	0.999360		1098	0.720124	0.555097	0.042012	0.999440		449	0.822853	0.715915	0.097814	0.999440					
git_repositories	0.939998		1676	0.368642	0.814226	0.091916	0.939998		214	0.746463	0.857913	0.000325	0.956357					
loans	1.000000		1098	0.720124	0.555097	0.042012	1.000000		1924	0.593976	0.330661	0.003403	1.000000					
med_visits	0.798053		524	0.640090	0.728561	0.187901	0.798053		230	0.537250	0.471036	0.784007	0.798053					
stock_market	0.518937		892	0.429435	0.523481	0.006785	0.520889		1055	0.825380	0.246962	0.028924	0.520889					

Table 1: Wyniki dla podzbioru 25% danych: ROC AUC i Accuracy dla modeli globalnych, Random Search (RS) i Bayesian Optimization (BO) wraz z optymalnymi hiperparametrami.

dataset	Global						RS						BO					
	ROC AUC						Accuracy											
	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC	Accuracy	n_estimators	max_samples	max_features	min_samples_leaf	Accuracy	
bank_marketing	0.911631		892	0.429435	0.523481	0.006785	0.911631		1259	0.545920	0.763633	0.000122	0.928008					
frauds	1.000000		184	0.470731	0.780121	0.052568	1.000000		1172	0.641117	0.892837	0.020782	1.000000					
git_repositories	0.850147		892	0.429435	0.523481	0.006785	0.850147		913	0.948025	0.572694	0.000190	0.975505					
loans	1.000000		1098	0.720124	0.555097	0.042012	1.000000		1728	0.998848	0.448212	0.000925	1.000000					
med_visits	0.608657		892	0.429435	0.523481	0.006785	0.608657		86	0.933905	0.649065	0.010387	0.608673					
stock_market	0.530115		1098	0.720124	0.555097	0.042012	0.530228		1915	0.778938	0.119147	0.022174	0.531479					
	Accuracy																	
bank_marketing	0.900111		892	0.429435	0.523481	0.006785	0.900111		1438	0.421139	0.360619	0.000119	0.906038					
frauds	0.999240		541	0.413956	0.846941	0.050349	0.999360		449	0.822853	0.715915	0.097814	0.999600					
git_repositories	0.939998		524	0.640090	0.728561	0.187901	0.939998		214	0.746463	0.857913	0.000325	0.940951					
loans	1.000000		1098	0.720124	0.555097	0.042012	1.000000		1924	0.593976	0.330661	0.003403	1.000000					
med_visits	0.798075		1676	0.368642	0.814226	0.091916	0.798075		230	0.537250	0.471036	0.784007	0.798075					
stock_market	0.521802		1098	0.720124	0.555097	0.042012	0.523256		1055	0.825380	0.246962	0.028924	0.524502					

Table 2: Wyniki dla podzbioru 50% danych: ROC AUC i Accuracy dla modeli globalnych, Random Search (RS) i Bayesian Optimization (BO) wraz z optymalnymi hiperparametrami.

dataset	Global						RS						BO					
	ROC AUC						Accuracy											
	ROC AUC	n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC		n_estimators	max_samples	max_features	min_samples_leaf	ROC AUC		n_estimators	max_samples	max_features	min_samples_leaf	Accuracy
bank_marketing	0.912019		892	0.429435	0.523481	0.006785	0.912019											
frauds	1.000000		541	0.413956	0.846941	0.050349	0.999360											
git_repositories	0.850397		892	0.429435	0.523481	0.006785	0.850397											
loans	1.000000		1098	0.720124	0.555097	0.042012	1.000000											
med_visits	0.531702		892	0.429435	0.523481	0.006785	0.531702											
	Accuracy																	
bank_marketing	0.899168		892	0.429435	0.523481	0.006785	0.899168											
frauds	0.999067		892	0.429435	0.523481	0.006785	0.999067											
git_repositories	0.939998		1891	0.669866	0.967436	0.683065	0.939998											
loans	0.999991		1098	0.720124	0.555097	0.042012	0.999991											
stock_market	0.522536		541	0.413956	0.846941	0.050349	0.522536											

Table 3: Wyniki dla podzbioru 75% danych: ROC AUC i Accuracy dla modeli globalnych oraz Random Search (RS) wraz z optymalnymi hiperparametrami.

Za każdym razem dla podzbiorów oryginalnych danych najlepszą globalnie wyznaczoną kombinacją wartości hiperparametrów dla ROC AUC (Accuracy) okazało się:

$$\text{n_estimators} = 892, \text{max_samples} = 0.429435, \text{max_features} = 0.523481, \text{min_samples_leaf} = 0.006785.$$

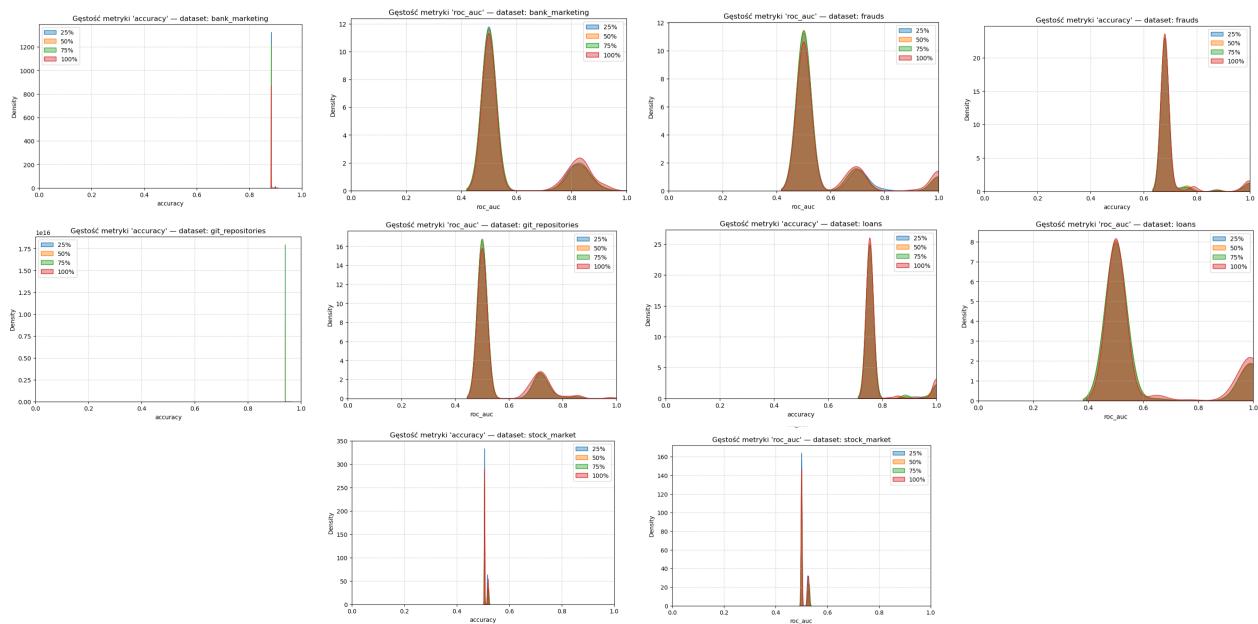
$$(\text{n_estimators} = 892, \text{max_samples} = 0.429435, \text{max_features} = 0.523481, \text{min_samples_leaf} = 0.006785)$$

Co ciekawe taki zestaw hiperparametrów dawał najlepsze wyniki na podzbiorach aż trzech zbiorów danych, zarówno rozpatrując roc auc jak i accuracy.

Test statystyczny Wilcoxon'a z hipotezami $H_0 : RS = BO, H_1 : RS < BO$ na poziomie istotności $\alpha = 0.05$ z podziałem na zbiory danych dla accuracy oraz roc auc wykonany na podzbiorach zawierających 25% oraz 50% oryginalnych rekordów skutkował odrzuceniem hipotezy zerowej w każdym przypadku, za wyjątkiem zbioru 'med_visits' przy 25% oryginalnych rekordów.

3.4.2 Wykresy

Rozkłady wyników dla podzbiorów uzyskane przy użyciu RS



Rozkłady wyników dla podzbiorów uzyskane przy użyciu BO

