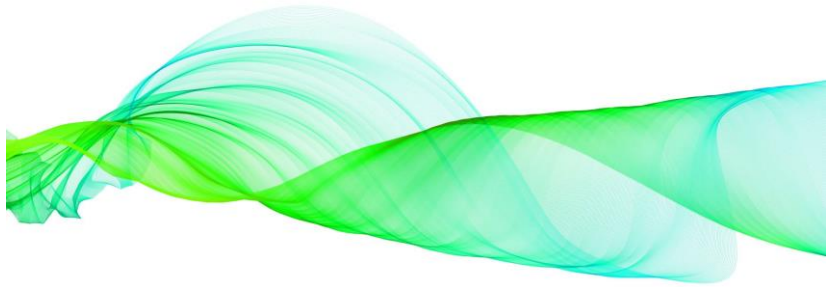
ENSEMBLE LEARNING

ALEKSANDRA SIEPIELA

MAJ, 2022 R.



Czym jest Ensemble learning?

Ensemble learning, czyli metody grupowania to technika uczenia maszynowego, która łączy kilka podstawowych modeli w celu stworzenia jednego optymalnego modelu predykcyjnego.

Podział metod grupowania

Metody grupowania można podzielić na dwie grupy:

- **sekwencyjne metody zespołowe**, w których zbiory uczące się są generowane sekwencyjnie. Podstawową motywacją metod sekwencyjnych jest wykorzystanie zależności między podstawowymi „uczniami”. Ogólna wydajność może zostać zwiększona poprzez ważenie wcześniej źle oznaczonych przykładów z wyższą wagą.
- **równoległe metody zespołowe**, w których zbiory uczące się podstawowe są generowane równolegle. Podstawową motywacją metod równoległych jest wykorzystywanie niezależności pomiędzy „uczniami” podstawowymi, ponieważ błąd można radykalnie zmniejszyć przez uśrednienie.

Metody Ensemble learning, które poznaliśmy dotychczas:

- Bagging

-Random forest

-Boosting

-Stacking

Warto również wspomnieć, że metody Bayesowskie dla regresji nieparametrycznej również możemy uznać za metody uczenia zespołowego.

Metody uczenia zespołowego w pierwszej kolejności wybierają populację pewnych bazowych „uczniów” na podstawie zbioru danych treningowych, a następnie łączą ich tworząc pewien złożony predyktor.

Na dzisiejszych zajęciach omówimy między innymi jednak technikę, która idzie o krok dalej. Jest to metoda boostingu, która buduje model zespołowy poprzez regularyzowane i nadzorowane wyszukiwanie w przestrzeni wielowymiarowej słabych „uczniów”.

PENALIZED REGRESSION

Intuicja

Intuicję dotyczącą powodzenia metody gradient boostingu ze ściąganiem możemy zdobyć poprzez nakreślenie pewnych analogii z metodą penalized linear regression, w przypadku, gdy rozpatrywana baza jest duża.

Penalized Regression

Rozważmy słownik wszystkich możliwych drzew regresyjnych $\mathcal{T} = \{\mathcal{T}_k\}$ z liczbą liści J. Drzewa te są realizowane na danych uczących jako funkcje bazowe w R^p. Rozważmy następujący model liniowy:

$$f(x) = \sum_{k=1}^{K} \alpha_k T_k(x), \qquad (16.1)$$

gdzie $K = card(\mathcal{T})$. Przyjmijmy, że współczynniki są estymowane przy pomocy metody najmniejszych kwadratów. W związku z tym, iż prawdopodobnie liczba takich drzew będzie dużo większa niż nawet największy zbiór danych uczących, wymagana jest pewna forma regularyzacji.

Penalized Regression c.d.

Niech $\hat{\alpha}(\lambda)$ będzie rozwiązaniem poniższego wyrażenia:

$$\min_{\alpha} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{k=1}^{K} \alpha_k T_k(x_i) \right)^2 + \lambda \cdot J(\alpha) \right\}, \qquad (16.2)$$

gdzie $J(\alpha)$ jest funkcją współczynników, która każe większe wartości. Przykładowo:

$$J(\alpha) = \sum_{k=1}^{K} |\alpha_k|^2 \qquad \text{ridge regression,} \qquad (16.3)$$

$$J(\alpha) = \sum_{k=1}^{K} |\alpha_k| \qquad \text{lasso,} \qquad (16.4)$$

Penalized Regression c.d.

Tak jak omawialiśmy już we wcześniejszych rozdziałach rozwiązanie problemu lasso z umiarkowanym do dużego λ jest zazwyczaj rzadkie, ponieważ wiele współczynników jest zerowanych ($\hat{\alpha}(\lambda)_k = 0$). W związku z tym tylko niewielka część wszystkich możliwych drzew wchodzi do modelu. Podejście to wydaje się rozsądne, ponieważ tylko niewielka liczba drzew będzie rzeczywiście istotna w aproksymacji dowolnej funkcji celu. Współczynniki, które nie są zerowane przez lasso, są ściągane do zera.

Forward stagewise strategy

Ze względu na bardzo dużą liczbę funkcji bazowych T_k rozwiązanie bezpośrednio (16.2) przy pomocy kary Lasso jest niemożliwe. Jednakże istnieje wykonalny algorytm - **Forward stagewise strategy**, któy dość dobrze przybliża efekt, który otrzymalibyśmy po zastosowaniu Lasso.

Warto zaznaczyć, że algorytm ten można wykorzystać dla dowolnego zbioru funkcji bazowych.

Forward stagewise strategy

Algorithm 16.1 *Forward Stagewise Linear Regression.*

1. Initialize $\check{\alpha}_k = 0,\ k = 1, \ldots, K$. Set $\varepsilon > 0$ to some small constant, and M large.

2. For $m = 1$ to M:

   (a) $(\beta^*, k^*) = \arg\min_{\beta,k} \sum_{i=1}^{N} \left(y_i - \sum_{l=1}^{K} \check{\alpha}_l T_l(x_i) - \beta T_k(x_i) \right)^2$.

   (b) $\check{\alpha}_{k^*} \leftarrow \check{\alpha}_{k^*} + \varepsilon \cdot \text{sign}(\beta^*)$.

3. Output $f_M(x) = \sum_{k=1}^{K} \check{\alpha}_k T_k(x)$.

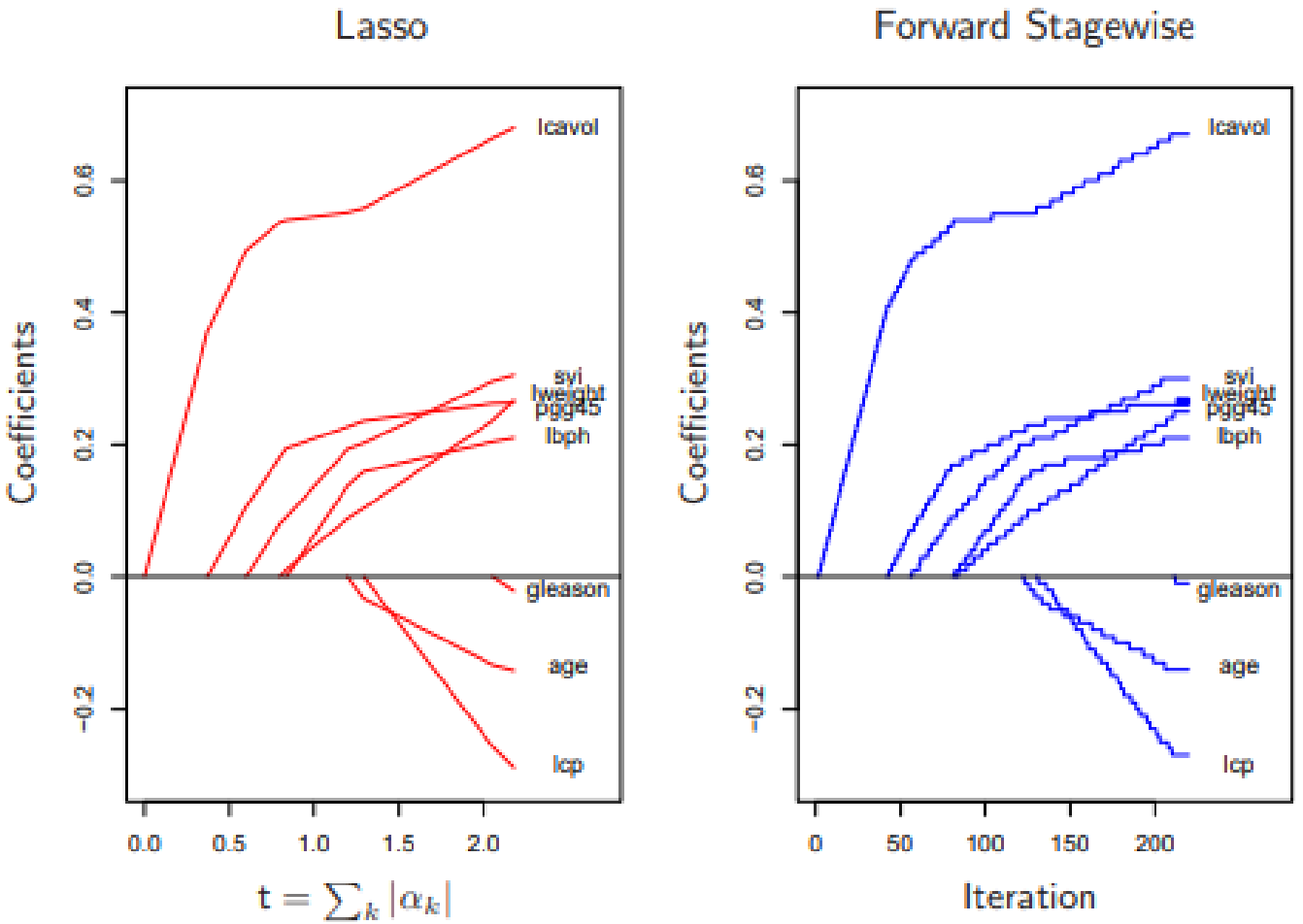


FIGURE 16.1. *Profiles of estimated coefficients from linear regression, for the prostate data studied in Chapter 3. The left panel shows the results from the lasso, for different values of the bound parameter* $t = \sum_k |\alpha_k|$. *The right panel shows the results of the stagewise linear regression Algorithm 16.1, using* $M = 220$ *consecutive steps of size* $\varepsilon = .01$.

Forward stagewise strategy a korelacje

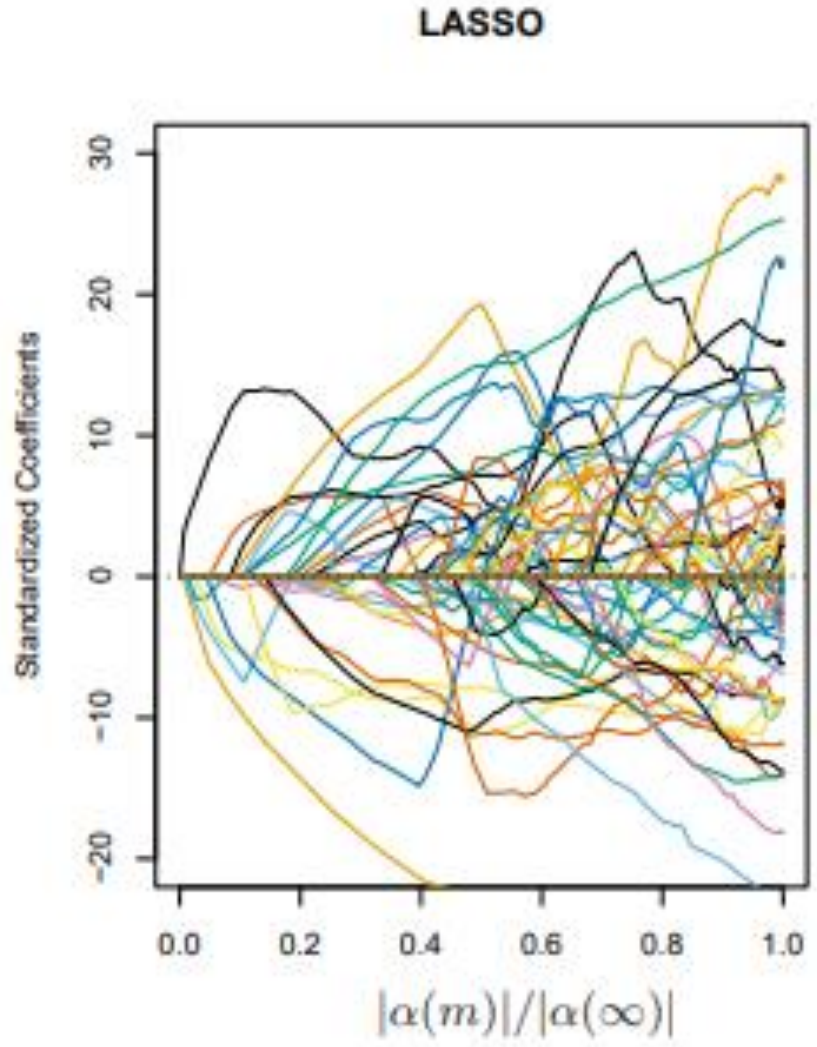
W niektóych sytuacjach podobieństwo otrzymanych wyników Lasso i Forward stagewise strategy jest uderzające. Na przykład jeżeli rozważamy bazowe funkcje T_k, które są wzajemnie nieskorelowane, to wówczas algorytm 16.1 daje dokładnie takie same wyniki jak lasso dla parametru ograniczającego $t = \sum_k |\alpha_k|$.

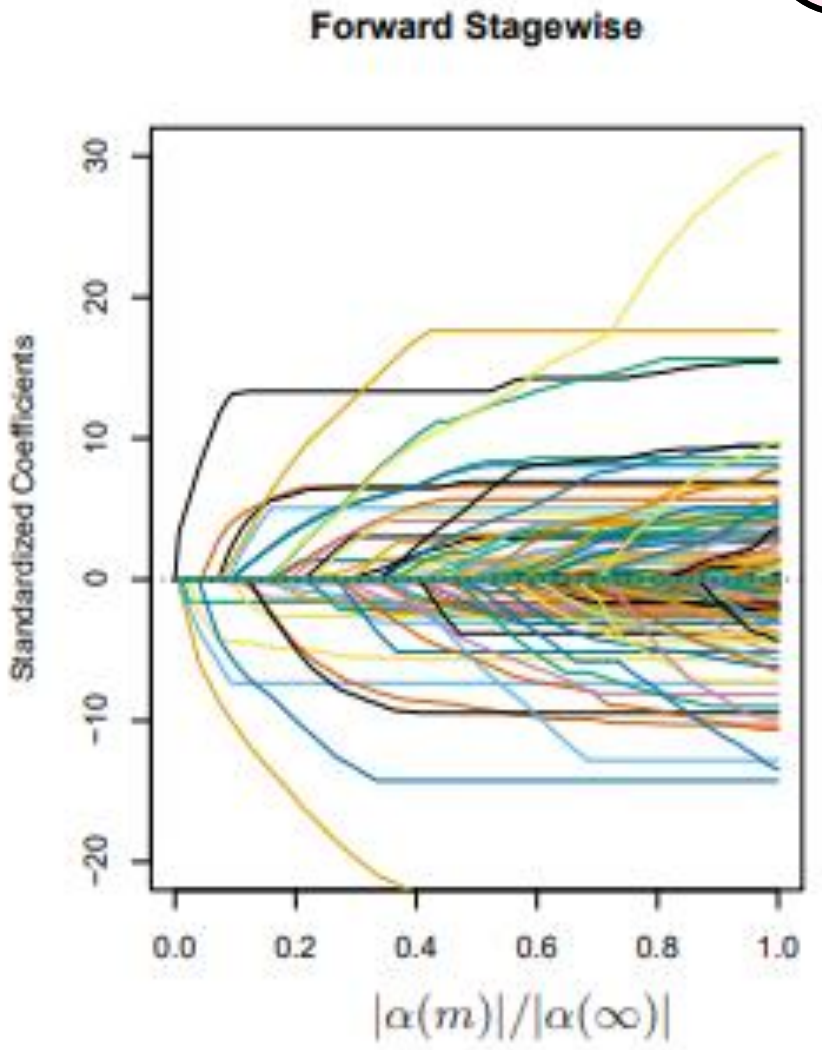
Forward stagewise strategy a korelacje c.d

Warto również wspomnieć, że jeżeli funkcjami bazowymi są drzewa to są one skorelowane. Zbiory rozwiązań dla Forward stagewise strategy są jednak identyczne jak dla Lasso, ale tylko jeśli wszystkie współczynniki $\hat{\alpha}_k(\lambda)$ są monotonicznymi funkcjami λ (co często się zdarza gdy korelacje pomiędzy zmiennymi są niskie). W innym przypadku zbiory rozwiązań nie są identyczne.

Algorytm Forward stagewise strategy przypomina boosting ze ściąganiem (tree boosting with shinkage) z parametrem szybkości uczenia ν odpowiadającym ϵ.

ŚCIEŻKI REGULARYZACJI


LASSO
Standardized Coefficients
30
20
10
0
−10
−20
0.0
0.2
0.4
0.6
0.8
1.0
$|\alpha(m)|/|\alpha(\infty)|$


Forward Stagewise
Standardized Coefficients
30
20
10
0
−10
−20
0.0
0.2
0.4
0.6
0.8
1.0
$|\alpha(m)|/|\alpha(\infty)|$

Działanie obu modeli jest dość podobne. Ponadto, osiągają one takie samo minimum. Metoda lasso szybciej zbytnio dopasowuje się do danych.

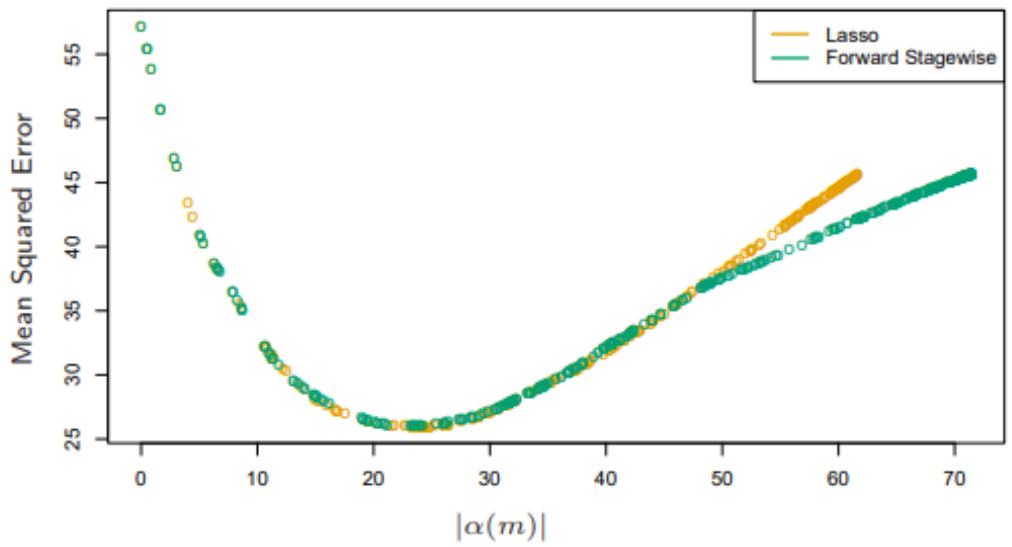


FIGURE 16.4. *Mean squared error for lasso and infinitesimal forward stagewise on the simulated data. Despite the difference in the coefficient paths, the two models perform similarly over the critical part of the regularization path. In the right tail, lasso appears to overfit more rapidly.*

LEARNING ENSEMBLES

Bardziej efektywny model zespołowy

Spostrzeżenia zdobyte w poprzednich sekcjach można wykorzystać do stworzenia bardziej efektywnego i wydajnego modelu zespołowego. Po raz kolejny rozpatrzmy funkcje następującej postaci:

$$f(x) = \alpha_0 + \sum_{T_k \in \mathcal{T}} \alpha_k T_k(x), \qquad (16.8)$$

gdzie $\mathcal{T}$ jest słownikiem funkcji bazowych, zazwyczaj drzew. W przypadku gradient boostingu i lasów losowych $|\mathcal{T}|$ jest bardzo duże i typowo modele te, finalnie, zawierają wiele tysięcy drzew.Zauważymy również, że gradient boosting ze ściąganiem dopasowuje się do monotonicznej ścieżki regularyzowanej normą L_1 w przestrzeni drzew.

Bardziej efektywny model zespołowy

W roku 2003 Friedman i Popescu zaproponowali pewne hybrydowe podejście, które można przedstawić w dwóch krokach:

- zadajemy skończony słownik $\mathcal{T}_L = T_1(x), T_2(x), ..., T_M(x)$ funkcji bazowych dla danych uczących;
- budujemy rodzinę funkcji $f_\lambda(x)$ poprzez dopasowanie ścieżki lasso w tym słowniku:

$$\alpha(\lambda) = \arg\min_{\alpha} \sum_{i=1}^{N} L[y_i, \alpha_0 + \sum_{m=1}^{M} \alpha_m T_m(x_i)] + \lambda \sum_{m=1}^{M} |\alpha_m|. \quad (16.9)$$

Bardziej efektywny model zespołowy

Rozważając najprostszą formę tego modelu, możemy stwierdzić, że jest to pewien post-procesing gradient boostingu lub lasów losowych, jeśli oczywiście przyjmiemy, że T_L jest zbiorem drzew wyprodukowanych odpowiednio przez algorytm gradient boostingu lub lasów losowych.

Poprzez dopasowanie ścieżki lasso do drzew, redukujemy zbiór, którego użyjemy, co wpływa na zredukowanie obliczeń i oszczędza pamięć naszego komputera.

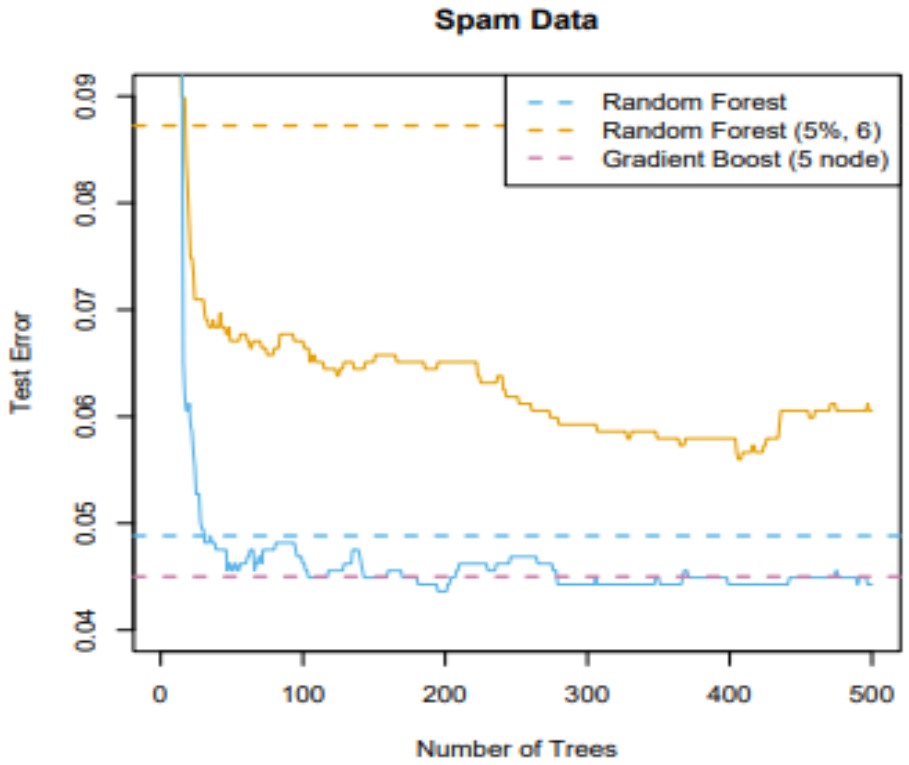


FIGURE 16.6. *Application of the lasso post-processing (16.9) to the spam data. The horizontal blue line is the test error of a random forest fit to the spam data, using* 1000 *trees grown to maximum depth (with* $m = 7$*; see Algorithm 15.1). The jagged blue curve is the test error after post-processing the first* 500 *trees using the lasso, as a function of the number of trees with nonzero coefficients. The orange curve/line use a modified form of random forest, where a random draw of* 5% *of the data are used to grow each tree, and the trees are forced to be shallow (typically six terminal nodes). Here the post-processing offers much greater improvement over the random forest that generated the ensemble.*

Wnioski z wykresu

Na wykresie obserwujemy

- Dla metody post-processing lasso otrzymujemy nieco lepsze wyniki niż dla lasu losowego (niebieska krzywa);
- Metoda post-processing lasso redukuje las do około 40 drzew z rozważanego na początku 1000;
- Działanie post-processed lasso jest porównywalne do gradient boostingu;
- Pomarańczowa krzywa reprezentuje zmodyfikowaną wersję lasu losowego, która została zaprojektowana w taki sposób, aby redukować korelacje pomiędzy drzewami w jeszcze większym stopniu.

DZIĘKUJĘ ZA UWAGĘ