POLITECHNIKA WARSZAWSKA
WYDZIAŁ MATEMATYKI
I NAUK INFORMACYJNYCH

PRACA DYPLOMOWA MAGISTERSKA

MATEMATYKA

Analiza metod uczenia maszynowego wykorzystywanych w budowaniu sygnatur genetycznych

Autor:
Katarzyna Sobiczewska

Promotor: dr hab. inż. Przemysław Biecek prof. nzw.

Warszawa, XII 2016

.....
podpis promotora

.....
podpis autora

Streszczenie

Tematyką niniejszej pracy jest porównanie metod uczenia maszynowego, używanych do budowy sygnatur genetycznych. Praca przybliża konstrukcje algorytmiczne dwóch procedur klasyfikacji: regresji logistycznej jako przykład uogólnionego modelu liniowego oraz nieliniowego podejścia, którego przykładem są lasy losowe. W odniesieniu do regresji logistycznej omówiona została również regularyzacja LASSO (L1). Opisane metody wykorzystano do analizy danych genomicznych, co pozwoliło na zbudowanie modelu określającego stopień macierzystości nowotworów złośliwych. W prezentowanej analizie mierzymy się również z problemem niezrównoważonych klas oraz przeuczeniem modelu.

Owocem przeprowadzonych badań, których wyniki zawarte są w prezentowanej pracy, jest narzędzie `RStemnessScorer`, pakiet do środowiska programistycznego R, pozwalający na ocenę macierzystości danej komórki na podstawie danych genomicznych.

Słowa kluczowe:

uczenie maszynowe, selekcja zmiennych, sygnatura genetyczna, ekspresja genów, regresja logistyczna, regularyzacja L1, lasy losowe, The Cancer Genome Atlas, Progenitor Cell Biology Consortium

The English version of a title:
Analysis of machine learning methods applied to building of genetic signatures

Abstract

The following thesis presents two machine learning techniques for building a gene signature: Logistic Regression as an example of a Generalized Linear Model and Random Forest as a non-linear approach. In the case of Logistic Regression L1 regularization was applied to select relevant features. Described methods are used on genomic data, which allows to create a statistical model of cancer stemness score. During the analysis we faced problems such as imbalanced labels and overfitting.

By the research, which is presented in this thesis in detail, I came with a tool that allows to score cancer's stemness based on its genomic profile: an R package `RStemnessScorer`.

Key words:
machine learning, feature selection, gene expression signature, genes expression, logistic regression, L1 regularization, Random Forest, The Cancer Genome Atlas, Progenitor Cell Biology Consortium

Spis treści

Rozdział 1

Wprowadzenie

Kiedy opracowano metody cięcia DNA na dowolne fragmenty
i łączenia ich ze sobą w zaplanowanej kolejności (...),
prawdopodobieństwo wykorzystywania całej tej wiedzy dla
dobra ludzkości stało się całkiem realne. Byłem wstrząśnięty.
Biologia zatem odznacza się matematyczną elegancją.
Życie ma sens.

Francis Collins

Kiedy w roku 1990 pod kierownictwem Francisa Collinsa ruszył ambitny, międzynarodowy Program Poznania Ludzkiego Genomu (ang. *Human Genom Project*), rozpoczęła się nowa era w dziedzinie genetyki i diagnozowania chorób. Działania podjęte na rzecz sekwencjonowania ludzkiego genomu przyczyniły się m. in. do poznania genotypu wirusów i podjęcia skuteczniejszego ich zwalczania, a także do identyfikacji mutacji odpowiadających różnym rodzajom nowotworów. Szczególnie to ostatnie dało światu nadzieję na zwycięstwo w nierównej jak dotąd walce człowieka z rakiem.

Poznanie pełnej sekwencji ludzkiego genomu ogłoszono 13 lat po rozpoczęciu badań, w 50. rocznicę odkrycia struktury podwójnej helisy DNA, skutkiem wspólnego wysiłku naukowców z USA, Chińskiej Republiki Ludowej, Francji, Niemiec, Japonii i Wielkiej Brytanii. Koniec Programu dał początek nowym wyzwaniom stawianym nie tylko genetykom i lekarzom, ale również specjalistom innych dziedzin, bez których nie byłoby możliwe pozyskiwanie, przechowywanie i przetwarzanie ogromnej ilości danych[1]. Szczególne znaczenie w tym miejscu - oprócz istotnego wkładu informatyki - ma praca

[1]Genom ludzki zawiera około 3 miliardów par zasad, które składają się na ok. 30 000 genów [14].

matematyków, których szeroki warsztat analityczny umożliwia skuteczną eksplorację peta-bajtowej otchłani danych w poszukiwaniu wiedzy, pozwalającej ekspertom podejmować lepsze decyzje (za: [9]). Niezastąpionym narzędziem w tym warsztacie jest statystyka i związane z nią metody uczenia maszynowego.

W niniejszej pracy zmierzymy się z różnymi typami danych genomicznych pacjentów dotkniętych nowotworem oraz embrionalnych[2] komórek macierzystych, korzystając z dwóch podejść statystycznych: znanej od XIX wieku regresji logistycznej [1] (z regularyzacją LASSO) oraz stosunkowo młodej metody lasów losowych (2001) [2]. Obie pozwalają na stworzenie klasyfikatora binarnego czyli modelu nauczonego do podejmowania decyzji *tak* lub *nie* w oparciu o wyestymowane prawdopodobieństwo zajścia zdarzenia *tak*. Wykorzystując tę cechę wspomnianych metod spróbujemy odpowiedzieć na pytanie *czy* dana komórka jest macierzysta? W jakim stopniu?

Na pracę składają się trzy części. Pierwszą jest omówienie teoretycznego zaplecza wykorzystywanych metod uczenia maszynowego. Opisane w niej zostały zasady działania i własności obu modeli wykorzystując do tego celu dostępny matematyczny aparat. Druga część zawiera biologiczne wprowadzenie w tematykę przeprowadzonych badań oraz prezentację uzyskanych wyników. Podstawowym narzędziem używanym w tej części pracy jest otwarte środowisko obliczeniowe *R* [15] oraz stworzony na cele tej pracy pakiet `RStemnessScorer`. Trzecia część poświęcona jest dyskusji porównującej działanie obu metod uczenia maszynowego w kontekście odpowiedzi na pytanie o stopień macierzystości danych genomicznych. Całość podsumowana jest kodem źródłowym napisanym w środowisku *R*, pozwalającym na odtworzenie umieszonych wyników i grafik.

[2]Właściwie indukowanych pluripotencjalnych komórek macierzystych, o czym szerzej na kolejnych stronach tej pracy.

Rozdział 2

Metody statystyczne

Zadaniem klasyfikacji jest znalezienie odwzorowania (czy też klasyfikatora) $t\colon \mathbf{R}^p \to \mathcal{Y}$, a więc zależności pomiędzy zmiennymi objaśniającym a zmienną objaśnianą. Gdy zbiór wartości jest zbiorem dwuelementowym mamy do czynienia z tzw. klasyfikacją dwuetykietową.

Rozważmy n elementową próbę uczącą $L = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ złożoną z obserwacji wektora losowego $(\mathbf{X}, Y)$. Wektor $\mathbf{X} = (X^{(1)}, \ldots, X^{(p)}) \in \mathbb{R}^p$ zawiera p zmiennych objaśniających (inaczej: *cech* lub *atrybutów*) oraz Y jest zmienną objaśnianą taką, że $Y : \{\mathcal{G}, \mathcal{G}^c\} \to \mathcal{Y} = \{1, 0\}$ tzn. że zmienna Y należy do klasy $\mathcal{G}$ (wówczas $Y = 1$) versus nie należy do niej ($Y = 0$).

Można spodziewać się, iż różne cechy w różnym stopniu powiązane są ze zmienną y: jedne niosą istotną siłę predykcyjną, podczas gdy inne znacznie słabszą lub są zupełnie bez znaczenia. Celem tej pracy jest znalezienie klasyfikatora z przestrzeni o zredukowanej liczbie wymiarów $t\colon \mathbf{R}^q \to \mathcal{Y}$, gdzie $q \ll p$, zbudowanego na zbiorze najistotniejszych atrybutów. W dalszej części pracy wyselekcjonowany q-wymiarowy zbiór cech nazywany będzie *sygnaturą genetyczną*.

W niniejszym rozdziale omówię szerzej następujące dwie metody klasyfikacji: regresję logistyczną z regularyzacją LASSO (ang. *Least Absolute Shrinkage and Selection Operator*) - jako przykład liniowej (poprzez odpowiednią funkcję łączącą) metody oraz alternatywne podejście: lasy losowe czyli komitet nieliniowych klasyfikatorów: drzew decyzyjnych. Konstrukcja każdej z wymienionych metod w naturalny sposób prowadzi do selekcji naistotniejszych zmiennych, tym samym wyboru odpowiedniej sygnatury.

Model regresji logistycznej

Przynależność zmiennej objaśnianej Y do grupy $\mathcal{G}$ przy zadanym wektorze zmiennych objaśniających $\mathbf{X}$ wyrazić można prawdopodobieństwem $P(\mathbf{X}) = \mathbf{P}(Y = 1|\mathbf{X})$. Dzięki temu możemy spojrzeć na Y jako na zmienną losową o warunkowym rozkładzie dwupunktowym: $Y|\mathbf{X} \sim Bin(P(\mathbf{X}))$.

Parametr p rozkładu dwupunktowego zależy nie tylko od wektora losowego $\mathbf{X}$, ale również od jego współczynników liniowych $\boldsymbol{\beta}$. W parametryzacji liniowej modelujemy zatem prawdopodobieństwo $\pi(\boldsymbol{\beta}, \mathbf{X})$ zadane jak niżej:

$$\text{logit}(\pi(\boldsymbol{\beta}, \mathbf{X})) = \alpha + \sum_{g=1}^{p} \beta_g X^{(g)} \stackrel{\text{ozn.}}{=} \eta(\boldsymbol{\beta}, \mathbf{X}),$$

gdzie $\boldsymbol{\beta} = (\alpha, \beta_1, \ldots, \beta_p) \in \mathbb{R}^{p+1}$. Interesujące nas prawdopodobieństwo otrzymujemy z odwrotności funkcji logitowej, tzn:

$$\pi(\boldsymbol{\beta}, \mathbf{X}) = \text{logit}^{-1}(\pi(\boldsymbol{\beta}, \mathbf{X})) = \frac{e^{\eta(\boldsymbol{\beta}, \mathbf{X})}}{1 + e^{\eta(\boldsymbol{\beta}, \mathbf{X})}}. \tag{2.1}$$

Decyzja klasyfikacyjna dla konkretnej realizacji $\mathbf{x}_i$ zmiennej losowej $\mathbf{X}$ zależy od wartości jaką przyjmuje funkcja $\pi(\boldsymbol{\beta}, \mathbf{X})$, a zatem wprost zależy od wybranego wektora współczynników liniowych $\boldsymbol{\beta}$. Przyjmujemy, że gdy przy zadanych współczynnikach liniowych $\hat{\boldsymbol{\beta}}$ znana jest wartość π_1 i wynosi ona $\pi_1 \geq 0.5$ obserwacja $(\mathbf{x}_1, y_1)$ przynależy do grupy $\mathcal{G}$ (czyli $y_1 = 1$) w przeciwnym przypadku należy do $\mathcal{G}^c$ (i $y_1 = 0$). Podjęcie takiej decyzji dla znanych obserwacji $\mathbf{x}_1, \ldots, \mathbf{x}_n$ o nieznanych etykietach $y_1, \ldots, y_n$ zależy więc od wyestymowanego na zbiorze uczącym wektora $\hat{\boldsymbol{\beta}}$. Znalezienie odpowiedniej wartości $\hat{\boldsymbol{\beta}}$ jest dla nas najważniejszym zadaniem w modelu regresji logistycznej.

W dalszych rozważaniach przez następujące oznaczenia rozumiemy:

$\mathbf{x}_i, y_i$	- i-tą realizację odpowiednio wektora losowego $\mathbf{X} \in \mathbb{R}^p$ i zmiennej losowej Y,
$X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T \in \mathbb{R}^{n \times p}$	- macierz n różnych obserwacji wektora losowego $\mathbf{X}$.
$X' = (\mathbf{1}, X) \in \mathbf{R}^{n \times (p+1)}$	- macierz X z kolumną jedynek, dodaną na pierwszej pozycji,
$\mathbf{x}_i'$	- i-ty wiersz macierzy X',
x_{ij}	- j-ty atrybut i-tej realizacji, przy czym $x_{i,0} = 1$ dla każdego $i = 1, 2, \ldots, n$,
$\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$	- wektor obserwacji zmiennych losowych $Y_1 Y_2, \ldots, Y_n$.

Funkcja wiarogodności

Funkcją wiarogodności nazywana jest gęstość łączna n-elementowego wektora zmiennych losowych $(Y_1, \ldots, Y_n)$ w funkcji parametrów odpowiedniego rozkładu wyznaczony w punkcie $(y_1, \ldots, y_n)$. W naszym przypadku mamy do czynienia z próbą prostą, toteż rozkład łączny jest produktem jednowymiarowych rozkładów zmiennej $Y|\mathbf{X}$.

Załóżmy, że dane są obserwacje $\mathbf{y}$ wektora $(Y_1, \ldots, Y_n)$ niezależnych zmiennych losowych oraz $Y_i|\mathbf{X}_i \sim Bin(P(\mathbf{X}_i))$ oraz $\mathbf{x}_i$ są kolejnymi realizacjami wektorów losowych $\mathbf{X}_i$. Wówczas gęstość zmiennej losowej Y_i pod warunkiem $\mathbf{X}_i = \mathbf{x}_i$ wyraża się jako:

$$f_{Y_i|\mathbf{X}_i=\mathbf{x}_i}(y) = P(\mathbf{x}_i)^y (1 - P(\mathbf{x}_i))^{(1-y)} = \left(\frac{P(\mathbf{x}_i)}{1 - P(\mathbf{x}_i)}\right)^y (1 - P(\mathbf{x}_i)), \tag{2.2}$$

co w liniowej parametryzacji $\eta(\boldsymbol{\beta}, \mathbf{x}_i)$ oraz dzięki wyrażeniu (2.1) pozwala na zapis w funkcji parametru $\boldsymbol{\beta}$ w poniższy sposób:

$$f_{Y_i|\mathbf{X}_i=\mathbf{x}_i}(y, \boldsymbol{\beta}) = \frac{\left(e^{\eta(\boldsymbol{\beta},\mathbf{x}_i)}\right)^y}{1 + e^{\eta(\boldsymbol{\beta},\mathbf{x}_i)}}.$$

Aspekt losowości $\mathbf{X}$ nie jest dla nas szczególnie interesujący: przyjmujemy, że $\mathbf{x}_i$ są z góry zadane dla każdego $i = 1, 2, \ldots, n$, toteż celem uproszczenia zapisu oznaczmy: $\eta_i(\boldsymbol{\beta}) \overset{\text{ozn.}}{=} \eta(\boldsymbol{\beta}, \mathbf{x}_i) = \alpha + \sum_{g=1}^{p} \beta_g x_{ig}$ oraz $\pi(\boldsymbol{\beta}, \mathbf{x}_i) \overset{\text{ozn.}}{=} \pi_i(\boldsymbol{\beta})$, zachowując w pamięci fakt, iż obie funkcje mają miejsce przy znanym $\mathbf{x}_i$. Funkcja wiarogodności, jako produkt powyżej przedstawionych gęstości w punkcie $\mathbf{y} = (y_1, \ldots, y_n)$ jest więc następującą funkcją szukanego parametru $\boldsymbol{\beta}$:

$$L(\boldsymbol{\beta}|y_1, \ldots, y_n) = \prod_{i=1}^{n} \frac{\left(e^{\eta_i(\boldsymbol{\beta})}\right)^{y_i}}{1 + e^{\eta_i(\boldsymbol{\beta})}}.$$

Bez utraty ogólności przejdziemy do logarytmicznej transformacji powyższej funkcji.[1] Oznaczmy przez $l(\boldsymbol{\beta})$ log-wiarogodność w punkcie $\boldsymbol{\beta}$ przy danych realizacjach $y_1, y_2, \ldots, y_n$. Wówczas:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \eta_i(\boldsymbol{\beta}) - \log(1 + e^{\eta_i(\boldsymbol{\beta})}) \right] = \mathbf{y}^T \boldsymbol{\eta}(\boldsymbol{\beta}) - \sum_{i=1}^{n} \log(1 + e^{\eta_i(\boldsymbol{\beta})}), \tag{2.3}$$

gdzie $\boldsymbol{\eta}(\boldsymbol{\beta}) = (\eta_1(\boldsymbol{\beta}), \ldots, \eta_n(\boldsymbol{\beta}))^T$. W tak postawionym problemie wyznaczenie maksimum przez znalezienie miejsca zerowego pochodnej $l'(\boldsymbol{\beta})$ nie jest możliwe w sposób analityczny. Istnieją jednakże alternatywne metody. Jedna z nich polega na przybliżeniu danej funkcji szeregiem Taylora w zadanym punkcie $\tilde{\boldsymbol{\beta}} = (\tilde{\alpha}, \tilde{\beta}_1, \ldots, \tilde{\beta}_p)^T$ rozumianym jako jako bieżące przybliżenie parametru $\boldsymbol{\beta}$. Taki zabieg pozwala na reprezentację $l(\boldsymbol{\beta})$ w postaci metody ważonych najmniejszych kwadratów. Szczegóły tej procedury zostały opisane poniżej.

Funkcję $l(\boldsymbol{\beta})$ w rozwinięciu Taylora wokół punktu $\tilde{\boldsymbol{\beta}}$ przybliża się następującym wzorem:

$$l(\boldsymbol{\beta}) \approx l(\tilde{\boldsymbol{\beta}}) + \left(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\right)^T \boldsymbol{l}'(\boldsymbol{\beta}) + \frac{1}{2} \left(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\right)^T \boldsymbol{l}''(\tilde{\boldsymbol{\beta}}) \left(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\right) \tag{2.4}$$

Z równania (2.3) oraz zależności (2.1) otrzymujemy następujące postaci pierwszej i drugiej pochodnej funkcji $l(\boldsymbol{\beta})$ w punkcie $\tilde{\boldsymbol{\beta}}$:

$$\begin{aligned}
&\boldsymbol{l}'(\tilde{\boldsymbol{\beta}}) \overset{\text{ozn.}}{=} \tfrac{\partial}{\partial \boldsymbol{\beta}} l(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = \left(\mathbf{y} - \boldsymbol{\pi}(\tilde{\boldsymbol{\beta}})\right)^T X' && \text{(gradient)} \\
&\boldsymbol{l}''(\tilde{\boldsymbol{\beta}}) \overset{\text{ozn.}}{=} \tfrac{\partial^2}{\partial^2 \boldsymbol{\beta}} l(\tilde{\boldsymbol{\beta}})|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = -(X')^T W\ X' && \text{(macierz Hessego)}
\end{aligned}$$

[1] Zwróćmy uwagę, iż zagadnienie maksymalizacji funkcji wiarogodności L jest równoważne zagadnieniu maksymalizacji każdego monotonicznego przekształcenia $g(\cdot)$ funkcji L. Przekształcenie $g(\cdot) = \log(\cdot)$ jest szczególnym przypadkiem funkcji zachowującej monotoniczność, często wybieranym ze względu na uproszczenie obliczeń.

przy czym:

- $\boldsymbol{\pi}(\tilde{\boldsymbol{\beta}})$ jest n-elementowym wektorem wartości $\Big(\pi_1(\tilde{\boldsymbol{\beta}}), \ldots, \pi_n(\tilde{\boldsymbol{\beta}})\Big)^T$,
- $W = \operatorname{diag}\Big(\pi_i(\tilde{\boldsymbol{\beta}})\big(1 - \pi_i(\tilde{\boldsymbol{\beta}})\big)\Big)$.

Podstawiając znalezione pochodne do (2.4) uzyskujemy postać formy kwadratowej funkcji $l(\boldsymbol{\beta})$:

$$l_Q(\boldsymbol{\beta}) = -\frac{1}{2}\sum_{i=1}^{n} w_i \left(\frac{y_i - \pi_i(\tilde{\boldsymbol{\beta}})}{w_i} - \mathbf{x}_i'(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\right)^2 + c_1(\tilde{\boldsymbol{\beta}})^2, \tag{2.5}$$

przy czym ostatni człon jest wyrazem stałym i nie ma znaczenia w procesie optymalizacji $l_Q(\boldsymbol{\beta})$. Powyższa postać pozwala nam przedstawić funkcję log-wiarogodności parametrów $\boldsymbol{\beta}$ w terminach zagadnienia ważonych najmniejszych kwadratów:

$$l_Q(\boldsymbol{\beta}) = -\frac{1}{2}\sum_{i=1}^{n} w_i \left(z_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2 + c_2(\tilde{\boldsymbol{\beta}})^2, \tag{2.6}$$

gdzie:

$$\begin{aligned} z_i &= \mathbf{x}_i'\tilde{\boldsymbol{\beta}} + \frac{y_i - \pi(\mathbf{x}_i, \tilde{\boldsymbol{\beta}})}{w_i} && \text{(ang. } \textit{working response}\text{)}, \\ w_i &= \pi(\mathbf{x}_i, \tilde{\boldsymbol{\beta}})\big(1 - \pi(\mathbf{x}_i, \tilde{\boldsymbol{\beta}})\big) && \text{(wagi)}. \end{aligned}$$

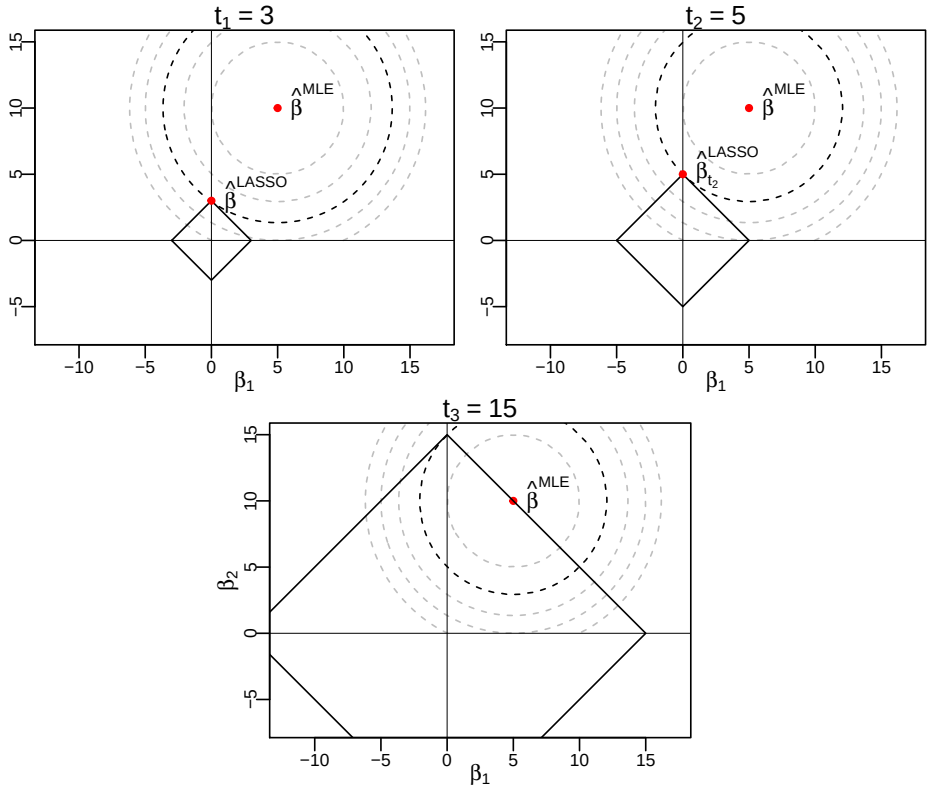
Ponadto $c_1(\tilde{\boldsymbol{\beta}})$ i $c_2(\tilde{\boldsymbol{\beta}})$ są odpowiednio zadanymi stałymi. Znalezienie argumentu maksimum funkcji $l_Q(\boldsymbol{\beta})$ pozwala na wyznaczenie estymatora parametru $\boldsymbol{\beta}$ zdefiniowanego na początku rozdziału.

Estymacja współczynników z regularyzacją LASSO

Tradycyjnym podejściem estymacji współczynników jest znalezienie argumentu maksimum funkcji (2.6) (estymatory największej wiarogodności). Celem redukcji wariancji estymatora $\hat{\boldsymbol{\beta}} = (\hat{\alpha}, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ można wprowadzić regularyzację, tzw. karę za zbyt duże współczynniki, tym samym przyczyniając się do zwiększenia obciążenia tego estymatora. Znane są regularyzacje o normach z przestrzeni ℓ_1 (regularyzacja LASSO), ℓ_2 (regresja grzbietowa), lub innych mieszanych normach (sieci elastyczne). Tak znalezione estymatory $\hat{\boldsymbol{\beta}}$ (obciążone, ale o zredukowanej wariancji) nazywamy *ściągającymi*. W tej pracy rozważę regularyzację LASSO.

Aby znalezione współczynniki $\hat{\boldsymbol{\beta}}$ były porównywalne w tej samej przestrzeni ℓ_1, należy zwrócić szczególną uwagę na efekt różnego skalowania poszczególnych zmiennych. Wyrugowanie takiego zjawiska następuje poprzez standaryzację każdej z kolumn macierzy obserwacji X. My zakładamy, że wszystkie z p atrybutów naszej macierzy obserwacji X są wyrażone na tej samej skali i nie wymagają standaryzacji.

Idea: Estymator LASSO współczynników $\boldsymbol{\beta}$ wyznacza się optymalizując zadaną funkcję celu przy ograniczeniu nałożonym na współczynniki. Optymalizacja przebiega po wszystkich współczynnikach $\beta_1, \ldots, \beta_p$ z pominięciem wyrazu wolnego α.



Rysunek 2.1: *Estymacja współczynników metodą LASSO na przykładzie dwóch atrybutów ($p = 2$). Okręgi są poziomicami funkcjonału $l_Q((\beta_1, \beta_2))$ w zależności od wartości tych współczynników. Estymator metody największej wiarogodności (MLE - ang.* Maximum Likelihood Estimator*), leżący w maksimum funkcjonału, tworzą niezerowe współrzędne. Optymalizacja funkcji celu $l_Q(\cdot)$ na kuli w ℓ_1 o promieniu t pozwala wyznaczyć taką rodzinę estymatorów $\hat{\beta}_t^{LASSO}$ (zależną od promienia tej kuli), by jedna ze współrzędnych była zerowa. Maksimum funkcjonału $l_Q(\cdot)$ na kuli o promieniu $t_1 = 3$ znajduje się na trzeciej poziomicy, zwiększenie promienia do $t_2 = 5$ obejmie również drugą poziomicę, dając tym samym większą wartość $l_Q(\cdot)$ w punkcie $\hat{\beta}_{t_2}^{LASSO}$. Dla $t_3 = 15$ estymator LASSO jest w istocie estymatorem największej wiarogodności. Zerowy promień kuli zeruje wszystkie współczynniki*

W naszym zagadnieniu funkcją celu jest log-wiarogodność, zatem interesuje nas jej maksymalizacja. Idea tej estymacji dla dwuwymiarowego wektora (β_1, β_2) przedstawiona jest na rysunku 2.1. W zależności od parametru ograniczającego t możemy wyznaczyć całą rodzinę estymatorów: $S_t(\hat{\beta}^{\text{LASSO}})$:

$$S_t(\hat{\beta}^{\text{LASSO}}) = \underset{\beta_1,\ldots,\beta_p}{\arg\max}\{l_Q(\beta_1, \ldots, \beta_p)\} \quad \text{przy ograniczeniu} \quad \sum_{g=1}^{n} |\beta_g| \leq t, \tag{2.7}$$

lub równoważnie w postaci Lagrange'a:

$$S_\lambda(\hat{\beta}^{\text{LASSO}}) = \underset{\beta_1,\ldots,\beta_p}{\arg\max}\{l_Q(\beta_1, \ldots, \beta_p) + \lambda \sum_{g=1}^{p} |\beta_g|\}. \tag{2.8}$$

Parametry λ (kary) oraz t (promienia) są ze sobą monotonicznie związane ($t = f(\lambda)$) i kontrolują wielkość ściągania: czym większa (mniejsza) jest wartość λ (t) tym większa jest wielkość ściągająca estymator $\hat{b}$, tzn. tym mocniej współczynniki ściągane są w kierunku zera.

Naturą estymacji metodą LASSO jest ściąganie wielkości współczynników do zera, co można osiągnąć, o ile wielkość parametru t jest odpowiednio mała. Ta własność pozwala na selekcję atrybutów bardziej istotnych, poprzez wyzerowanie atrybutów mniej istotnych. Jeśli t wybierzemy większe niż $t_0 = \sum_{g=1}^{p} |\hat{\beta}_g|$, wówczas estymator LASSO jest w istocie ekstremum funkcji celu (czyli estymatorem MLE), a poziom ściągania jest zerowy (patrz: rysunek trzeci na ryc. 2.1). Biorąc np. $t = t_0/2$, estymator największej wiarogodności jest ściągany średnio o 50%. Dokładna natura ściągania nie jest oczywista i omówimy ją szerzej w następnym paragrafie.

Algorytm spadku po współrzędnych (ang. *coordinate descent*))

Istnieje kilka algorytmów estymacji współczynników metodą LASSO: szczególnie znany jest algorytm LAR [5], jednak w praca [11] pokazuje, że algorytm *coordinate descent* działa znaczenie szybciej. Jest to również algorytm zaimplementowany w funkcji `glmnet()` pakietu `R`, którą wykorzystuję w późniejszych symulacjach, co podwójnie uzasadnia omówienie tego podejścia w niniejszej pracy.

Implementacja algorytmu omawiana jest zazwyczaj ([11], [8]) na tradycyjnie rozumianym modelu liniowym. Poniżej omówię to podejście w sytuacji regresji logistycznej.

Algorytm *coordinate descent* dla regresji logistycznej

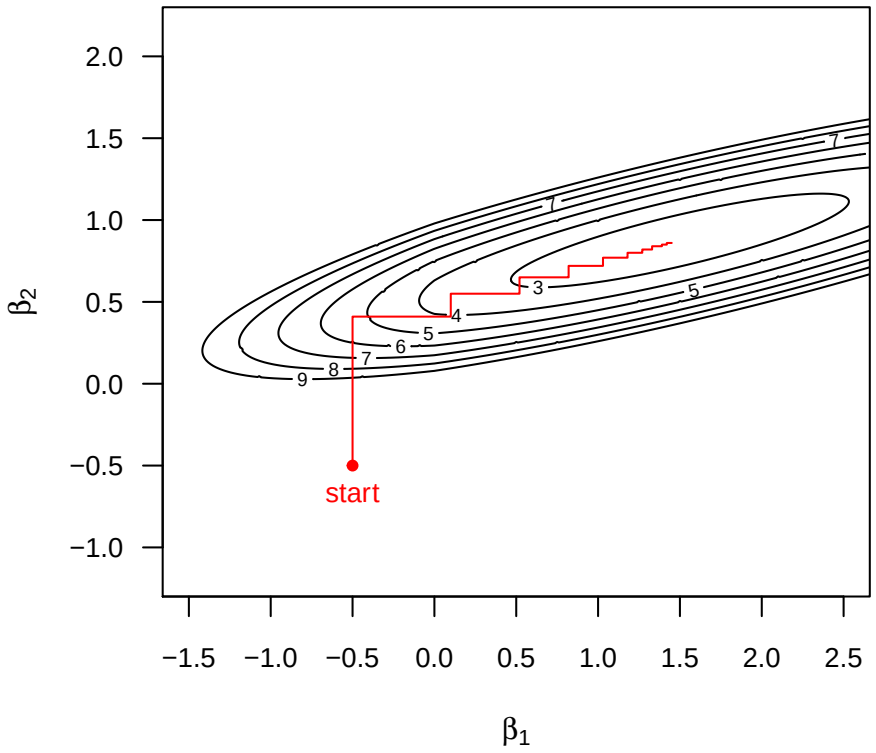
Zadaniem algorytmu jest sukcesywna estymacja każdego z parametrów, tzn. w kroku $k+1$ algorytmu wyznaczamy kolejno oszacowanie każdego z $j \in \{1, \ldots, p\}$ współczynników w następujący sposób:

$$\beta_j^{k+1} = \arg\max_v f(\beta_1^{k+1}, \ldots, \beta_{j-1}^{k+1}, v, \beta_{j+1}^{k}, \ldots, \beta_p^{k}) \tag{2.9}$$

do uzyskania zbieżności. Dla ustalonego λ rozważmy k+1–szy krok algorytmu. Z k-tego kroku otrzymaliśmy wartości: $\tilde{\boldsymbol{\beta}}^k = (\tilde{\alpha}^k,\ \tilde{\beta}_1^k,\ \ldots,\ \tilde{\beta}_p^k)^T$. Otrzymane estymatory pozwalają policzyć macierz W^k (wag) i wektor $\boldsymbol{z}^k$ (odpowiedzi roboczych):

$$z_i^k = \mathbf{x}_i' \tilde{\boldsymbol{\beta}}^k + \frac{y_i - \pi_i(\tilde{\boldsymbol{\beta}}^k)}{w_i^k},$$
$$w_i^k = \pi_i(\tilde{\boldsymbol{\beta}}^k)\big(1 - \pi_i(\tilde{\boldsymbol{\beta}}^k)\big).$$

dla wszystkich $i = 1, 2, \ldots, n$.



Rysunek 2.2: *Działanie algorytmu spadku po współrzędnych w sytuacji dwuwymiarowej. W każdym kroku algorytmu tylko jedna współrzędna jest optymalizowana*

Przy ustalonym parametrze λ i wartościach $\tilde{\boldsymbol{\beta}}^k$ funkcję celu z równania (2.8) możemy zapisać w postaci:

$$R_\lambda(\tilde{\alpha}^k, \tilde{\beta}_1^k, \ldots, \tilde{\beta}_j^k, \ldots, \tilde{\beta}_p^k) = \\ = -\frac{1}{2}\sum_{i=1}^{n} w_i^k \left(z_i^k - \tilde{\alpha}^k - \sum_{l \neq j} x_{il}\tilde{\beta}_l^k - \tilde{\beta}_j^k x_{ij} \right)^2 + \lambda \sum_{l \neq j} |\tilde{\beta}_l^k| + \lambda |\tilde{\beta}_j^k|.$$

Wyznaczenie wartości β_j^{k+1} dla kolejnych j sprowadza się (na podstawie (2.9)) do policzenia miejsca zerowego pochodnej cząstkowej $\frac{\partial R_\lambda}{\partial v}$ na j-tej współrzędnej wektora $(\beta_1, \ldots, \beta_p)$, tzn. $\frac{\partial}{\partial v} R_\lambda^k(v) \overset{\text{ozn.}}{=} \frac{\partial R_\lambda}{\partial v}(\tilde{\alpha}^k, \tilde{\beta}_1^k, \ldots, \tilde{\beta}_{j-1}^k, v, \tilde{\beta}_{j+1}^k, \ldots, \tilde{\beta}_p^k)$:

$$\frac{\partial}{\partial v} R_\lambda^k(v) = \sum_{i=1}^{n} w_i^k x_{ij} \left(z_i^k - \tilde{\alpha}^k - \sum_{l \neq j} x_{il}\tilde{\beta}_l^k - v x_{ij} \right) + \lambda \ \mathrm{sgn}(v) \qquad (2.10)$$

(o ile zmienna v jest niezerowa). Zauważmy, że wyrażenie $z_i^k - \tilde{\alpha}^k - \sum_{l \neq j} x_{il}\tilde{\beta}_l^k$ odpowiada resztom częściowym zmiennej *working response*, co zapiszemy jako: $z_i^k - \eta_i^{(j)}(\tilde{\boldsymbol{\beta}}^k)$ i możemy ją traktować jako iteracyjną zmienną odpowiedzi. Powyższa równość w postaci

wektorowej przyjmuje postać:

$$\begin{aligned}\frac{\partial}{\partial v}R_\lambda^k(v) &= X^{(j)T}W^k\big(\boldsymbol{z}^k - \boldsymbol{\eta}^{(j)}(\tilde{\boldsymbol{\beta}}^k)\big) - X^{(j)T}W^kX^{(j)}v + \lambda\ \mathrm{sgn}(v) = \\ &= X^{(j)T}W^kX^{(j)}\left(\left(X^{(j)T}W^kX^{(j)}\right)^{-1}X^{(j)T}W^k\big(\boldsymbol{z}^k - \boldsymbol{\eta}^{(j)}(\tilde{\boldsymbol{\beta}}^k)\big) - v\right) + \lambda\ \mathrm{sgn}(v) = \\ &= X^{(j)T}W^kX^{(j)}\left(\hat{\beta}_j^{WMNK} - v\right) + \lambda\ \mathrm{sgn}(v)\end{aligned}$$

Czynnik $X^{(j)T}W^kX^{(j)}$ jest dodatni (poza trywialnym przypadkiem, gdy j-ta kolumna macierzy eksperymentu jest zerowa). Zauważmy, że odjemna w powyższym zapisie jest estymatorem ważonej metody NK dla parametru β_j przy zmiennej wyjaśnianej będącej częściową resztą iteracyjnej zmiennej odpowiedzi, ściślej: $\boldsymbol{z}^k - \boldsymbol{\eta}^{(j)}(\tilde{\boldsymbol{\beta}}^k)$.

Powyższe równanie posiada dokładne rozwiązanie, w rezultacie otrzymujemy nowe przypisanie:

$$\beta_j^{k+1} \leftarrow \mathcal{S}\left(\hat{\beta}_j^{WMNK}, \left(X^{(j)T}W^kX^{(j)}\right)^{-1}\lambda\right), \tag{2.11}$$

gdzie funkcja:

$$\mathcal{S}(\beta,\gamma) = \mathrm{sgn}(\beta)(|\beta| - \gamma)_+ = \begin{cases} \beta - \gamma & \text{gdy} \quad \beta > 0 \quad \text{oraz} \quad \gamma < |\beta|\,, \\ \beta + \gamma & \text{gdy} \quad \beta < 0 \quad \text{oraz} \quad \gamma < |\beta|\,, \\ 0 & \text{gdy} \quad \gamma \geq |\beta| \end{cases} \tag{2.12}$$

jest nazywana [11] operatorem *soft-threshold* (ryc. 2.3). Kluczową jego własnością jest „ściąganie" do zera argumentów β, będących odpowiednio małymi.

Ponadto równanie (2.10) pozwala zauważyć, iż algorytm ten sukcesywnie „wygasza" kolejne współczynniki modelu. Innymi słowy współczynnik β_j, który został ściągnięty do zera w k-tym kroku algorytmu, pozostaje zerowy we wszystkich kolejnych krokach.

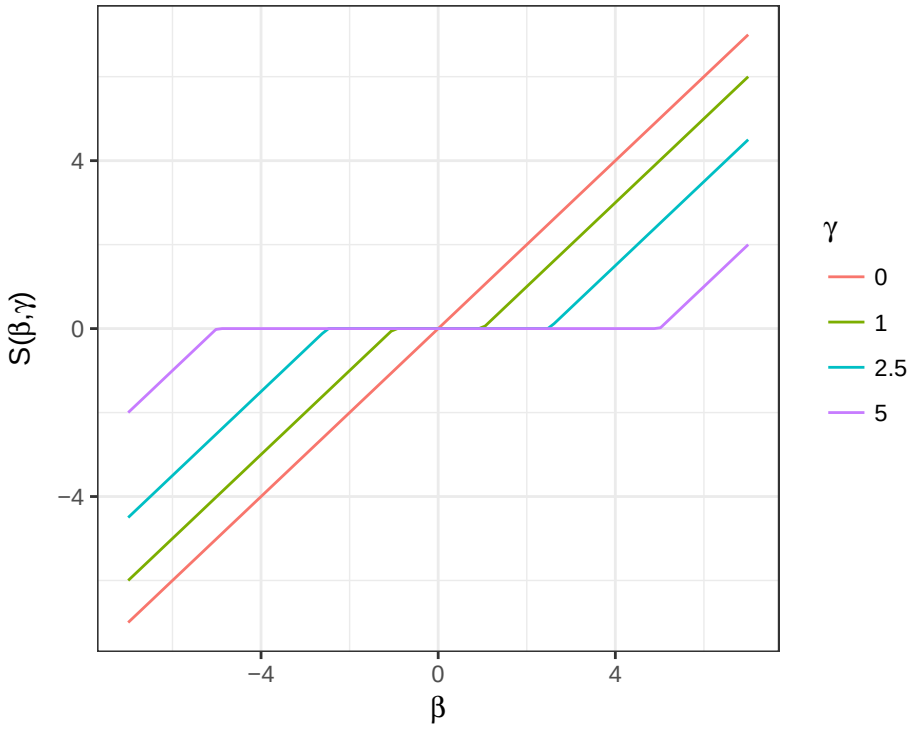
Estymacja α

Wyraz wolny wyznaczany jest analogicznie przez przyrównanie do zera pochodnej cząstkowej $\frac{\partial}{\partial\alpha}R_\lambda$. Ze względu na standaryzację macierzy eksperymentu, współczynnik α^{k+1} zadany jest w uproszczonej formie:

$$\alpha^{k+1} \leftarrow \tilde{z}^k = \tilde{\alpha}^k + \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \pi_i(\tilde{\boldsymbol{\beta}}^k)}{\pi_i(\tilde{\boldsymbol{\beta}}^k)(1 - \pi_i(\tilde{\boldsymbol{\beta}}^k))}.$$

Ścieżki współczynników

Ze względu na niski koszt obliczeniowy, opisana powyżej procedura pozwala efektywnie znaleźć rozwiązania na całej siatce lambd $(\lambda_{\min}, \ldots, \lambda_m, \ldots \lambda_{\max})$. Startujemy wówczas z pewnej maksymalnej wartości kary $\lambda_{\max}$, dla której wszystkie współczynniki są zerowe i powtarzamy cały proces (2.11) po każdej ze zmiennych do uzyskania zbieżności.



Rysunek 2.3: *Wykres operatora* soft-threshold. *Dla* $\gamma = 0$ *otrzymujemy referencyjną prostą* $S(\beta, \gamma) = \beta$. *Rosnące wartości kary* γ *oddają intuicję siły ściągania współczynników do zera*

Sukcesywnie zmniejszając λ_m powtarzamy procedurę, a ostatnio wyznaczone rozwiązania dla λ_{m+1} inicjują algorytm dla λ_m. Najogólniejszy schemat postępowania wygląda następująco:

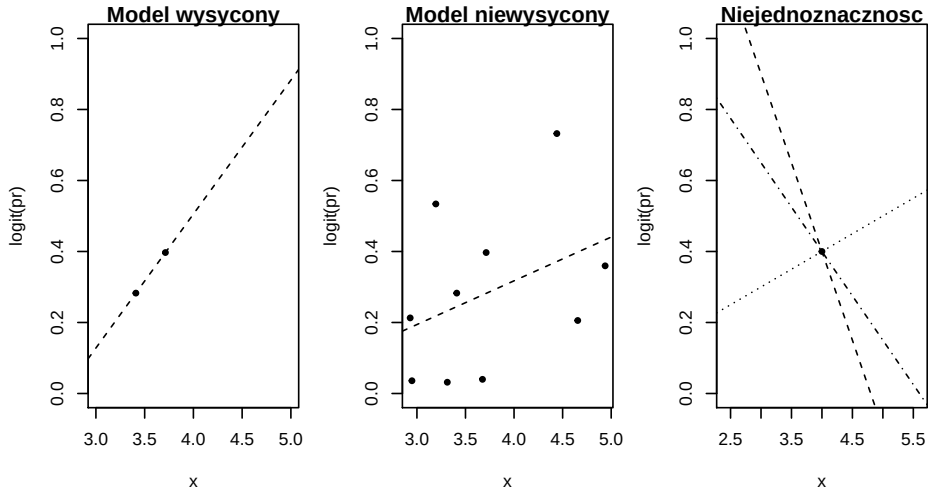
Algorytm 2.1. Wyznaczanie współczynników na siatce lambd

Zewnętrzna pętla: Dla kolejnych λ_m z siatki $\lambda_{\max}, \ldots, \lambda_{\min}$ wykonaj:

Środkowa pęta: używając bieżących oszacowań $\tilde{\boldsymbol{\beta}}^k$ wyznacz kwadratową aproksymację $l_Q(\cdot)$.

Wewnętrzna pętla: wykonaj algorytm spadku po współrzędnych na ważonej metodzie najmniejszych kwadratów z odpowiednią karą λ_m.

Uwaga: W sytuacji, gdy liczba estymowanych parametrów $p+1$ (wliczywszy wyraz wolny) jest większa niż liczba obserwacji n nie możemy przebiegać z wartościami λ do zera. Model wysycony nie jest zdefiniowany w takiej sytuacji (patrz rysunek 2.4). Zatem domyślnie malejącą sekwencję λ zamyka $\lambda_{\min} = \epsilon\lambda_{\max}$, gdzie ϵ jest takie, że dla wartości $\epsilon\lambda_{\max}$ wybrana liczba parametrów będzie odpowiadała modelowi wysyconemu.



Rysunek 2.4: *Ilustracja relacji liczby parametrów modelu do liczby danych obserwacji na przykładzie dwuparametrowego modelu regresji logistycznej ($p = 1$ plus wyraz wolny) w reprezentacji liniowej dla obserwacji $\mathbf{x}_i \in \mathbb{R}$. Pierwszy wykres ilustruje sytuację, gdy liczba parametrów jest taka sama jak liczba obserwacji, tzn. $p + 1 = n$ (model* wysycony*). Na drugim wykresie widzimy sytuację modelu* niewysyconego*: liczba obserwacji jest większa, niż wymiar problemu ($n > p + 1$). Ostatni wykres jest ilustracją sytuacji przeciwnej: liczba obserwacji jest mniejsza, niż wymiar modelu ($n < p+1$). W takim przypadku model nie mógł zostać wyznaczony jednoznacznie*

Prezentacja wybranych funkcji celu

Do tej pory jedyną optymalizowaną przez nas miarą była funkcja log-wiarogodności $L(\boldsymbol{\beta}|\mathbf{y})$. Jednak wyznaczenie optymalnych parametrów $\boldsymbol{\beta}$ może mieć miejsce dzięki innym funkcjom celu. W poniższym paragrafie opisze kilka wybranych.

Dla porządku załóżmy, że przy ustalonym parametrze kary λ_m do modelu zostało włączonych q_m zmiennych, przy czym $q_m + 1 \leq n$ (aby model mógł być określony jednoznacznie). Oznaczymy:

Ω - model wysycony	– do modelu włączono q_Ω zmiennych, których liczba **odpowiada liczbie obserwacji** $n - 1$,
ω - model niewysycony	– do modelu włączono q_ω zmiennych, których liczba jest **mniejsza od liczby obserwacji** $n - 1$.

Dewiancja modelu

Często używaną miarą do oceny jakości dopasowania modelu o mniejszej liczbie parametrów jest odniesienie do modelu wysyconego. Taka „odległość" modelu ω od Ω nazywana jest *dewiancją*, a do jej wyznaczenia używa się poniższego wzoru:

$$dev_{\omega,\Omega} = -2 \log \frac{L(\omega|\mathbf{y})}{L(\Omega|\mathbf{y})}. \tag{2.13}$$

Przez optymalną wartość dewiancji rozumiemy najmniejszą możliwą.

W modelu wysyconym wartości dopasowane $\hat{\mathbf{y}}^{\Omega}$ (czyli decyzje podjęte na podstawie odpowiednich $\hat{\boldsymbol{\beta}}^{\Omega}$, wybranych do modelu Ω) są tożsame dostępnym etykietom $\mathbf{y} = y_1, \ldots, y_n$. To oznacza, że $P(\mathbf{x}_i) = \hat{y}_i^{\Omega} = y_i$ (patrz: przypis[2]). Podstawiając to do wzoru (2.2) zauważymy, że:

$$L(\Omega|\mathbf{y}) = \prod_{i=1}^{n} f_{Y_i|\mathbf{X}_i=\mathbf{x}_i}(y_i) = \prod_{i=1}^{n} P(\mathbf{x}_i)^{y_i}(1-P(\mathbf{x}_i))^{(1-y_i)} = \prod_{i=1}^{n} \left(\hat{y}_i^{\Omega}\right)^{y_i} \left(1-\left(\hat{y}_i^{\Omega}\right)\right)^{(1-y_i)} = 1.$$

Analogiczna wartość $L(\omega|\mathbf{y})$ nie jest tak oczywista do policzenia, ponieważ $P(\mathbf{x}_i)$ nie przyjmują skrajnych wartości 0 lub 1 i wyznacza się je poprzez $\pi_\omega \overset{\text{ozn.}}{=} \pi(\hat{\boldsymbol{\beta}}^{\omega}, \mathbf{x}_i)$:

$$L(\omega|\mathbf{y}) = \prod_{i=1}^{n} \pi_\omega^{y_i} \left(1-\pi_\omega\right)^{(1-y_i)}.$$

Uwzględniając powyższe uwagi, postać dewiancji upraszcza się i w logarytmicznej reprezentacji wiarogodności wynosi:

$$dev_{\omega,\Omega} = -2l(\omega|\mathbf{y}) = -2\sum_{i=1}^{n} y_i \log(\pi_\omega) + (1-y_i)\log(1-\pi_\omega).$$

Im bardziej dewiancja bliższa wartości 0, tym trafniejsze mamy predykcje π_ω.

Procent poprawnej klasyfikacji

Znacznie bardziej intuicyjnym narzędziem jest pomiar frakcji poprawnych klasyfikacji (ACC, ang. *accuracy*). Jest to procentowa wartość, którą naturalnie chcielibyśmy maksymalizować:

$$ACC_\omega = \frac{\sum_{i=1}^{n} \mathbb{I}(\hat{y}_i^{\omega} = y_i)}{n} \cdot 100\%.$$

Powyżej przedstawiona funkcja celu swą intuicyjność okupuje znaczną stratą informacji. Decyzję o wartości $\hat{y}_i^{\omega}$ podejmuje się ustalając odgórnie pewien próg odcięcia, najczęściej $k = 0.5$. Wówczas $\hat{y}_i^{\omega} = 1$, gdy $\pi_\omega \geq k$ i 0 w przeciwnym przypadku. Tym, co tracimy, jest informacja o odległości poszczególnych wartości $\pi(\hat{\boldsymbol{\beta}}^{\omega}, \mathbf{x}_i)$ od przyjętego progu. Również zadana z góry wartość k może nie mieć uzasadnienia w konkretnym problemie, zwłaszcza gdy błędne oszacowanie $\hat{y}_i^{\omega} = 1$, podczas gdy $y_i = 0$ jest kosztowniejsze niż błędne oszacowanie $\hat{y}_i^{\omega} = 0$, gdy $y_i = 1$ (lub odwrotnie).

Powierzchnia pod krzywą ROC (AUC)

Zasadniczym celem klasyfikacji jest poprawne wskazanie wyróżnionej klasy $\mathcal{G}$ (TP, ang. *true positive*) oraz prawidłowe niewskazanie drugiej klasy (TN, ang. *true negative*). Kiedy niepoprawnie wskazujemy na wyróżnioną klasę (FP) lub nie wskazujemy wyróżnionej

[2]Model dopasowany jest **dokładnie** do dostępnych danych, czyli prawdopodobieństwo $P(\mathbf{x}_i)$, że i-ta obserwacja należy do grupy $\mathcal{G}$ wynosi dokładnie 1 lub 0, czyli odpowiada wartości y_i.

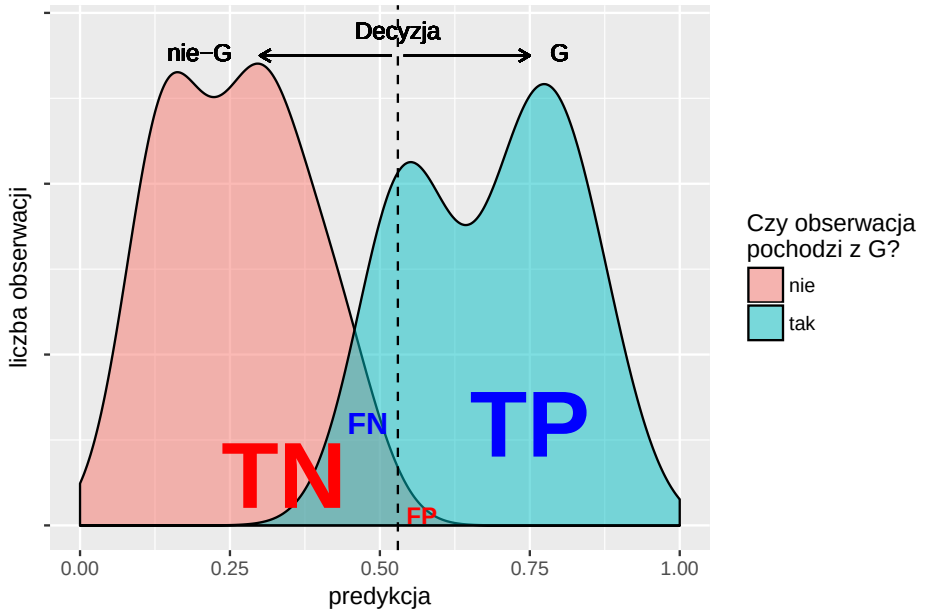
klasy, gdy powinniśmy (FN), popełniamy błąd klasyfikacji[3]. Zauważmy przy okazji, że $ACC_\omega = \frac{TP+TN}{TP+TN+FN+FP}$.

	Przewidziano $\mathcal{G}$	Przewidziano $\mathcal{G}^c$
Zaobserwowano $\mathcal{G}$ $(y_i = 1)$	TP $= \sum_{i=1}^n \mathbb{I}(\hat{y}_i^\omega = y_i)$	FN $= \sum_{i=1}^n \mathbb{I}(\hat{y}_i^\omega \neq y_i)$
Zaobserwowano $\mathcal{G}^c$ $(y_i = 0)$	FP $= \sum_{i=1}^n \mathbb{I}(\hat{y}_i^\omega \neq y_i)$	TN $= \sum_{i=1}^n \mathbb{I}(\hat{y}_i^\omega = y_i)$

Tabela 2.1: *Tabelka klasyfikacyjna: porównanie wyników modelu ze stanem rzeczywistym*

Korzystając z powyżej przyjętych oznaczeń wprowadzimy dwie, często wykorzystywane miary oceny klasyfikacji:

$$\begin{aligned} \text{Czułość} &= TPR = \frac{TP}{TP+FN} \\ 1 - \text{Specyficzność} &= FPR = \frac{FP}{TN+FP} \end{aligned}$$

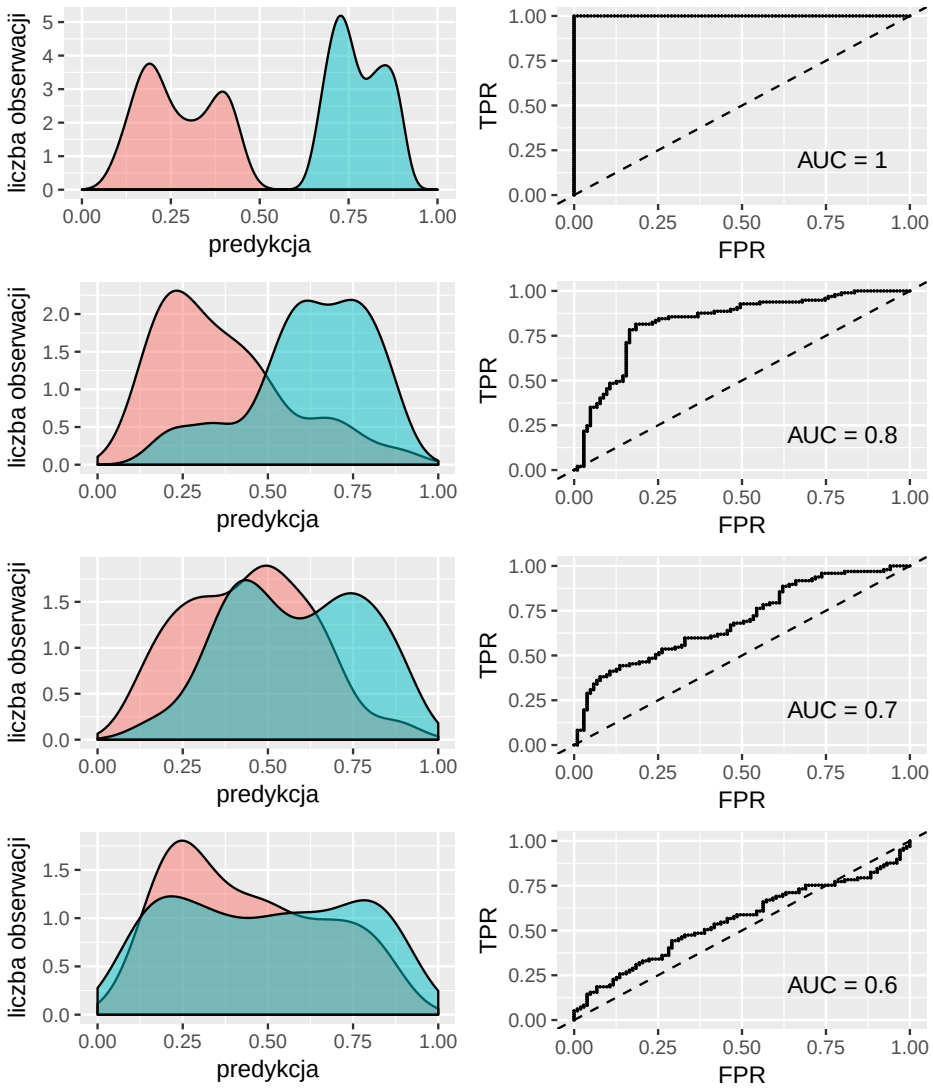


Rysunek 2.5: *Wizualizacja istoty miar TN, TP, FN i FP. Wykresy są funkcją liczby obserwacji od znalezionego predyktora w podziale na etykietę obserwacji. Punkt odcięcia k dla tej reguły decyzyjnej znajduje się w punkcie* 0.53. *Obserwacje o predyktorze* $\pi(\boldsymbol{\beta}^\omega, \mathbf{x}_i) \geq k$ *klasyfikowane są do klasy* $\mathcal{G}$. *Obserwacje z klasy* $\mathcal{G}$ *(niebieskie) na lewo od punktu odcięcia są fałszywie negatywne (FN). Obserwacje z klasy* $\mathcal{G}^c$ *(czerwone) na prawo od punktu odcięcia są fałszywie pozytywne (FP). Czułością nazwiemy stosunek obszaru TP do całego niebieskiego obszaru. Specyficzność to frakcja TN w odniesieniu do pola czerwonego. Przedstawiono jądrowy estymator gęstości*

Jeśli każdą z posortowanych wartości $\pi(\hat{\boldsymbol{\beta}}^\omega, \mathbf{x}_i)$, $i = 1, 2, \ldots, n$ potraktujemy jako punkt odcięcia, możemy stworzyć n reguł decyzyjnych i dla każdej z nich wyznaczyć

[3]Często błędy te nie mają tej samej wagi. Przez błąd pierwszego rodzaju α rozumiemy fałszywy „sygnał" danego testu, czyli (*false positive*). Błąd drugiego rodzaju β to niewykrycie przez test „sygnału", który wystąpił (*false negative*)

tabelki kontyngencji oraz miary czułości i specyficzności, do oceny danej reguły. Wykres rozrzutu tych miar dla kolejnych $\pi(\hat{\boldsymbol{\beta}}^{\omega}, \mathbf{x}_i)$ nazywany jest krzywą ROC. Celem jest wybranie takiego progu decyzyjnego k spośród $\{\pi(\hat{\boldsymbol{\beta}}^{\omega}, \mathbf{x}_i) \colon i = 1, 2, \ldots, n\}$, który najlepiej rozdzieli dwie klasy biorąc pod uwagę frakcje błędów pierwszego i drugiego rodzaju.



Rysunek 2.6: *Krzywa ROC - rozrzut miar czułości (TPR) i 1-specyficzności (FPR) dla reguł decyzyjnych, na przykładzie 4 różnych testów. Wskaźnikiem jakości klasyfikatora jest pole pod krzywą ROC: AUC (ang.* area under curve*). Górny wykres ilustruje wygląd krzywej ROC w sytuacji idealnej separacji dwóch klas. Coraz słabiej rozróżnialne klasy wypłaszczają krzywą ROC, by dla testu nierozróżniającego pomiędzy klasami sprowadzić ją do prostej referencyjnej $TPR = FPR$*

Każda z trzech miar jakości dopasowania modelu ma swoje wady i zalety. Najbardziej intuicyjną miarą jest ACC, jednak wiąże się ona z odgórnym ustaleniem progu odcięcia, powyżej którego podejmujemy decyzję pozytywną, a ten nie zawsze musi być

skutecznie wybrany. Optymalizowanie modelu pod kątem miary AUC eliminuje problem wyboru progu odcięcia. Należy jednak brać pod uwagę, iż AUC pozwala nam na uzyskanie predykcji dopasowanych *względem* siebie, nie zaś takich, które będą dążyły do granic przedziału $[0, 1]$. Obserwowanie dewiancji modelu, pozwala zmniejszać odchylenie predykcji od klasyfikacyjnej decyzji 0 lub 1.

Lasy losowe

Lasami losowymi (ang. *random forest*) nazywamy rodzinę wielu klasyfikatorów zbudowaną na różnych podpróbkach próby uczącej L, uzyskanych metodą bootstrapową. Do klasyfikacji obserwacji z każdej z podpróbek wykorzystuje się drzewa decyzyjne, dlatego zaczniemy od omówienia tej metody.

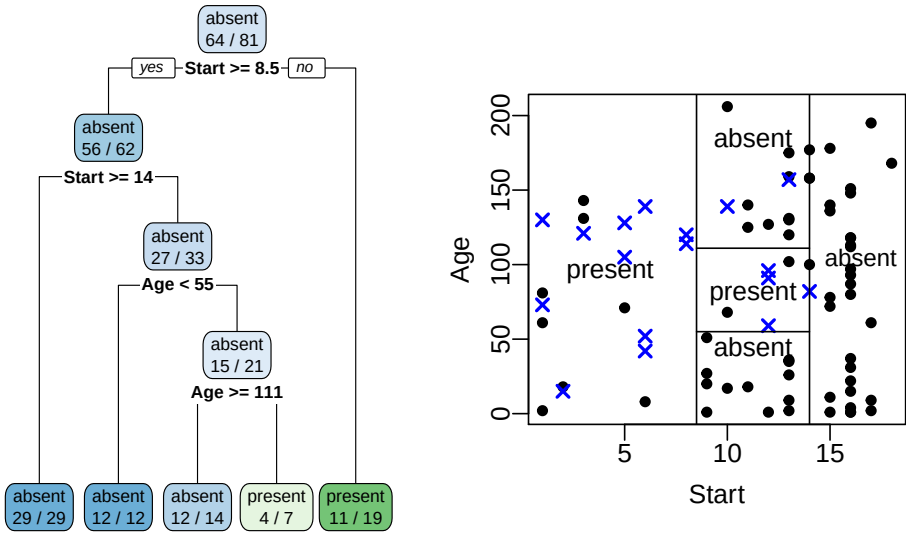
Drzewa decyzyjne

Drzewo w matematyce ma swoją szczególną definicję - taką nazwę przyjmuje nieskierowany acykliczny graf spójny. W problemie klasyfikacji rozważamy drzewa, których wierzchołek początkowy będziemy nazywali korzeniem. W przypadku drzew decyzyjnych wierzchołki nazywane są *węzłami*, zaś krawędzie: *gałęziami*. Jeśli z pewnego węzła wychodzą gałęzie, to jest on nazywany *rodzicem* zaś węzły pochodne zwane są *dziećmi*. Węzeł, który nie ma dzieci nazywany jest *liściem*. Z konstrukcji drzewa wynika, że do każdego liścia prowadzi jedna droga.

Nadal rozważamy próbę uczącą $L = \{(\mathbf{x}_i, y_i) \colon i = 1, \ldots, n\}$, gdzie $\mathbf{x}_i \in \mathbb{R}^p$ oraz $y_i \in \{0, 1\}$. Niech m oznacza dowolny, ustalony węzeł drzewa T. W każdym węźle, który nie jest liściem, następuje podział podpróby uczącej, która doń trafiła na dwie podgrupy ze względu na postawiony w tym węźle warunek na wybranym predyktorze. Ów podział wyznacza nam w p-wymiarowej przestrzeni wszystkich obserwacji hiperkostki czyli tzw. regiony decyzyjne. Reguły tego podziału są jednakowe dla wszystkich węzłów-rodziców i omówię je szczegółowo w dalszej części pracy. Ideę drzew decyzyjnych ilustruje przykład na rysunku 2.7.

Wyznaczanie obszarów decyzyjnych

Budowa drzewa decyzyjnego postępuje od korzenia, w którym znajduje się cała próba ucząca, poprzez kolejne węzły aż do liści, w których podejmowana jest decyzja. W każdym kolejnym węźle, poczynając od korzenia, należy wyznaczyć taki podział próby uczącej na dwie nowe podpróby w kolejnych węzłach, aby możliwie najlepiej zróżnicować dane obserwacje. Przez różnorodność klas w węźle rozumiemy rozkład klas w danej próbie, przy czym stosowna miara różnorodności powinna przyjmować wartość 0, gdy wszystkie obserwacje należą do tej samej klasy i maksymalną, gdy rozkład przynależności do klas jest jednostajny. Innymi słowy idealny podział próby w korzeniu na dwie



Rysunek 2.7: *Idea drzew decyzyjnych na przykładzie danych dotyczących wystąpienia pewnej choroby. Po lewej stronie widzimy przykładowe drzewo decyzyjne zbudowane na dwóch atrybutach: wieku* - Age *(w miesiącach) oraz liczbie pierwszych od góry licząc operowanych kręgów* - Start. *Drzewo posiada 9 węzłów, w każdym z 5 liści podejmowana jest decyzja większościowa (*present - *wystąpienie choroby,* absent - *w przeciwnym wypadku). Po prawej stronie widzimy odpowiedni podział dwuwymiarowej przestrzeni rozpiętej na atrybutach* Age *i* Start *na hiperkostki, wyznaczające właściwe obszary decyzyjne. Dane pochodzą z pakietu* **`rpart`** *do konstrukcji drzew decyzyjnych.*

nowe podpróby w pierwszym dziecku lokuje wszystkie obserwacje z jednej klasy, a w drugim z drugiej klasy.

Obserwacje, które znalazły się w m-tym węźle musiały spełnić w poprzednich węzłach pewne określone warunki, zatem wyznacza on w przestrzeni obserwacji $\mathcal{X}$ hiperkostkę R_m określoną tymi warunkami ($R_m \subset \mathcal{X}$). Niech liczba obserwacji, które znalazły się w węźle m wynosi n_m oraz

$$\hat{p}_{mk} = \frac{1}{n_m} \sum_{\mathbf{x}_i \in R_m} I(y_i = k)$$

będzie frakcją obserwacji z klasy k w obszarze R_m, $k = 0, 1$. Obserwacje w węźle m klasyfikujemy do klasy $k(m)$ zgodnie z *regułą maksymalnego prawdopodobieństwa*, tzn:

$$k(m) = \arg\max_k \hat{p}_{mk}.$$

Jeśli węzeł m jest liściem, to $k(m)$ wyznacza końcową klasyfikację wszystkich obserwacji $\mathbf{x}$ znajdujących się w obszarze R_m. W przeciwnym wypadku decyzja $k(m)$ informuje nas jedynie, która klasa jest najliczniej reprezentowana węźle m, a obserwacje z tego obszaru są dalej rozdzielane do poszczególnych węzłów-dzieci.

Załóżmy, że została ustalona pewna miara różnorodności $Q_m(T)$, jednakowa dla całego drzewa T. Niech m_L i m_R oznaczają dzieci węzła m (odpowiednio lewego i prawego) wyznaczone na podstawie progu a w cesze $x^{(g)}$. Innymi słowy do węzła m_R trafią obserwacje dla których wartości predyktora $x^{(g)}$ są większe od progu a, a do węzła m_L obserwacje o wartościach w $x^{(g)}$ mniejszych od a. Niech $\hat{p}_L$ jest frakcją elementów próby uczącej, które z węzła m zostały skierowane do węzła m_L:

$$\hat{p}_L = \frac{n_{m_L}}{n_m}.$$

Analogicznie definiujemy $\hat{p}_R$, a zatem $\hat{p}_R = 1 - \hat{p}_L$.

Jako łączną miarę różnorodności w węzłach-dzieciach przyjmujemy uśrednioną wartość różnorodności klas w dzieciach, tak by uśrednienie uwzględniało liczność obserwacji z próby uczącej w dzieciach, tj:

$$Q_{m_L,m_R}(T) = \hat{p}_L Q_{m_L}(T) + \hat{p}_R Q_{m_R}(T)$$

Miarą różnicy między różnorodnością klas w węźle m i w jego dzieciach nazywamy:

$$\Delta Q_{m,m_L,m_R}(T) = Q_m(T) - Q_{m_L,m_R}(T) \tag{2.14}$$

Istotą budowania drzewa jest znalezienie w każdym węźle

$$\arg\max_{x^{(g)},\ a} \Delta Q_{m,m_L,m_R}(T), \tag{2.15}$$

czyli takiego predyktora $x^{(g)}$ i takiego progu podziału a, aby zmaksymalizować różnorodność pomiędzy węzłem-rodzicem a węzłami-dziećmi.

Algorytm maksymalizacji $\Delta Q_{m,m_L,m_R}$

Dla ustalonego predyktora g (liczbowego lub kategorycznego), który w n-elementowej przyjmuje $W \leq n$ różnych wartości ze zbioru $V^g = \{v_1^g, \ldots, v_W^g\}$ mamy $\frac{1}{2}2^W - 1 = 2^{W-1} - 1$ możliwych podziałów tej próby na dwa rozłączne podzbiory[4]. To oznacza, że wybranie podziału na podstawie jednego wybranego predyktora wymaga policzenia miary $\Delta Q_{m,m_L,m_R}(T)$ $2^{W-1} - 1$ razy, co oczywiście przy nawet niedużej liczbie W staje się bardzo kosztowne.

Zauważmy jednak, że gdy predyktor g jest liczbowy lub ma wartości na skali porządkowej, można - korzystając z monotoniczności - łatwo ograniczyć liczbę podziałów do podziałów typu typu $x^{(g)} \leq v_i^g$ lub $x^{(g)} < v_i^g$. Ten prosty zabieg daje nam $W - 1$ możliwych podziałów.

[4]Liczba wszystkich możliwych podzbiorów na zbiorze W-elementowym wynosi 2^W. Wystarczy połowa z liczby 2^W, by wyznaczyć wszystkie możliwe podziały zbioru W-elementowego na dwa rozłączne podzbiory przy czym nie interesuje nas zbiór pusty.

Jeśli w zbiorze możliwych wartości V^g predyktora g nie ma naturalnego porządku (czyli gdy wartości są nominalne), możemy kierować się wartościami prawdopodobieństw warunkowych przynależności danej obserwacji do ustalonej klasy k pod warunkiem konkretnej wartości v_i^g. Przyjmijmy, że wartości v_i^g danego predyktora zostały ułożone według rosnących wartości $p(k|v_i^g)$:

$$p(1|v_1^g) \leq p(1|v_2^g) \leq \ldots \leq p(1|v_W^g).$$

Wówczas jeden z $W-1$ podziałów typu

$$\{v_1^g, \ldots, v_w^g\}, \quad \{v_{w+1}^g, \ldots, v_W^g\}$$

będzie maksymalizował wartość $\Delta Q_{m,m_L,m_R}(T)$.

Wybór miary różnorodności

Omówimy teraz, jaką miarę najlepiej wybrać do konstrukcji drzewa.

Najpopularniejszymi miarami różnorodności z węźle m danego drzewa T są: indeks Giniego oraz entropia. Inną, bardziej intuicyjną miarą, jest frakcja błędnych klasyfikacji. Jeśli za p przyjmiemy frakcję obserwacji w klasie $k=0$ (a tym samym $(1-p)$ oznacza frakcję obserwacji z klasy $k=1$), wówczas miary te zadane są odpowiednio następującymi wzorami:

$$Q_m(T) = \begin{cases} 2p(1-p) & \text{(indeks Giniego)} \\ -p\log p - (1-p)\log(1-p) & \text{(entropia)} \\ \min(p, 1-p) & \text{(frakcja błędnych klasyfikacji)} \end{cases}$$

zaś $Q_m(T)$ oznacza miarę różnorodności w węźle m danego drzewa T.

Rysunek 2.8 ilustruje możliwe miary różnorodności w węźle. Wszystkie są do siebie podobne, jednak wskaźnik Giniego i entropia są bardziej czułe na zmiany rozkładów klas niż frakcja błędnej klasyfikacji, stąd też te dwie metody są częściej wybierane. Poniższy przykład, inspirowany pozycją [8], lepiej uzasadnia taki wybór.

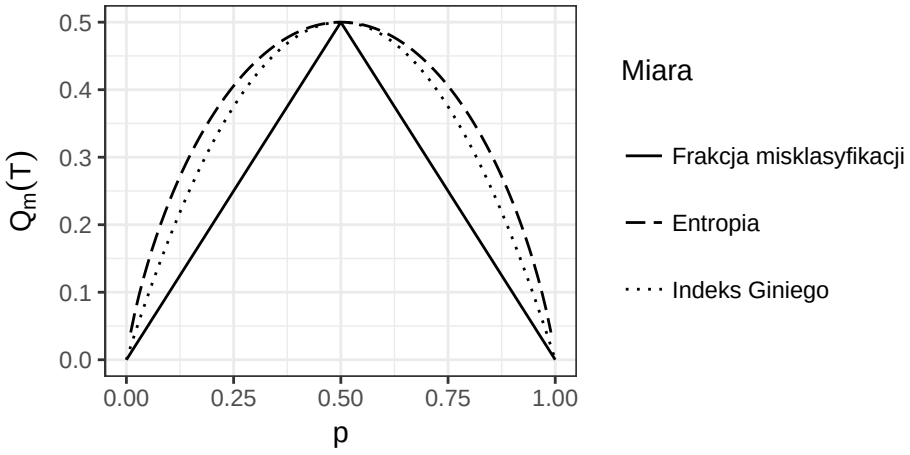
Przykład 2.2.1. Załóżmy, że w węźle rodzicu m znajduje się 800 obserwacji: 400 z klasy 1 i 400 obserwacji z klasy 2. Porównamy dwa różne podziały tego węzła (m_L i m_R są węzłami-dziećmi węzła m odpowiednio lewym i prawym):

A. $m_L = \begin{cases} \text{klasa 1: 300} \\ \text{klasa 2: 100} \end{cases}$ $m_R = \begin{cases} \text{klasa 1: 100} \\ \text{klasa 2: 300} \end{cases}$

B. $m_L = \begin{cases} \text{klasa 1: 200} \\ \text{klasa 2: 400} \end{cases}$ $m_R = \begin{cases} \text{klasa 1: 200} \\ \text{klasa 2: 0} \end{cases}$

Odpowiednie wyliczenia zebrane są w powyższej tabeli.

Intuicyjnie podział z punktu B. powinien być uznany jako lepszy, tzn. dający większe zmniejszenie różnorodności klas w węzłach niż podział A. Tymczasem frakcja błędnej klasyfikacji traktuje oba podziały w ten sam sposób, podczas gdy indeks Giniego i entropia zwracają większe zmniejszenie różnorodności w przypadku B niż w przypadku A.



Rysunek 2.8: *Trzy typy miar różnorodności $Q_m(T)$ w węźle m danego drzewa T. Miara entropii została przeskalowana tak, by jej maksimum również znalazło się w punkcie* $(0.5, 0.5)$.

	Frakcja błędnej klasyfikacji		Indeks Giniego		Entropia	
	A.	B.	A.	B.	A.	B.
$Q_m(T)$	$1/2$	$1/2$	$1/2$	$1/2$	≈ 0.69	≈ 0.69
$Q_{m_L}(T)$	$1/4$	$1/3$	$3/8$	$4/9$	≈ 0.56	≈ 0.64
$Q_{m_R}(T)$	$1/4$	0	$3/8$	0	≈ 0.56	0
$\hat{p}_L$	$1/2$	$3/4$	$1/2$	$3/4$	$1/2$	$3/4$
$\hat{p}_R$	$1/2$	$1/4$	$1/2$	$1/4$	$1/2$	$1/4$
$Q_{m_L,m_R}(T)$	$1/4$	$1/3$	$3/8$	$2/8$	≈ 0.56	≈ 0.48
$\Delta Q_{m,m_L,m_R}(T)$	**$1/4$**	**$1/4$**	**$1/8$**	**$1/6$**	**0.13**	**0.21**

Tabela 2.2: *Porównanie trzech miar różnorodności węzłów na podstawie przykładowego podziału 800 obserwacji.*

Istotność cech

Wpływ danej cechy X_k w drzewie T będziemy mierzyli poprzez miarę $Imp(X_k, T) = \sum_{t \in T} \Delta Gini(X_k, t)$, gdzie $\Delta Gini(X_k, t)$ oznacza zmniejszenie niejednorodności w węźle t, w którym atrybut X_k jest zmienną rozdzielającą, wyliczone z kryterium Giniego. Istotność cechy X_k (ang. *importance*) w modelu złożonym z m drzew zadana będzie poprzez $Imp(X_k) = \frac{1}{m} \sum_{j=1}^{m} Imp(X_k, T_j)$ (ang. *mean decrease Gini*) (ozn. MDG). Czym wyższa wartość $MDG(X_k)$ tym istotniejszy jest wpływ zmiennej X_k.

Lasy losowe czyli komitet drzew

Opisane powyżej drzewa decyzyjne posiadają wiele istotnych zalet, m.in. uniwersalność w obsłudze typów zmiennych, odporność na braki danych, intuicyjną interpretację mo-

delu a przede wszystkim możliwość uchwycenia złożonych interakcji pomiędzy zmiennymi. Należy jednak podkreślić, że metoda ta jest wysoce niestabilna. Ze względu na hierarchiczny charakter procedury małe zmiany próby uczącej w korzeniu przenoszą się do wszystkich następujących węzłów i mogą radykalnie zmienić postać modelu. Ta wada drzew nakazuje sięgać po stabilniejsze metody klasyfikacyjne. Budując klasyfikator złożony z rodziny drzew korzystamy ze wszystkich zalet drzew decyzyjnych jednocześnie eliminując ich największą wadę: wysoką wariancję.

Komitet klasyfikatorów

Istotą lasów losowych jest stabilizacja i poprawa własności predykcyjnych modeli możliwie dobrych i słabo od siebie zależnych. Mając bowiem wiele statystycznie niezależnych od siebie modeli, których prawdopodobieństwo podjęcia poprawnej decyzji jest większe od 1/2, to klasyfikując obserwację do klasy, na którą wskazała większość klasyfikatorów, dokonamy prawie na pewno poprawnej decyzji[5]. Intuicja podpowiada zatem, że prawdopodobieństwo popełnienia błędu przez las losowy będzie rosło, wraz z wzrostem odpowiednio zdefiniowanego współczynnika korelacji między pojedynczymi drzewami oraz malało wraz ze wzrostem siły predykcyjnej każdego z drzew.

Jedną ze znanych technik, umożliwiających redukcję wariancji poszczególnych klasyfikatorów jest *bagging* (ang. *bootstrap aggregation*). Załóżmy, że dysponujemy próbą treningową $\mathbf{Z} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, a naszym celem jest stworzenie modelu, zwracającego predykcję $\hat{f}(x)$ dla obserwacji $x = (x_1, \ldots, x_N)$. Wówczas na każdej z $b = 1, 2, \ldots, B$ prób bootstrapowych $\mathbf{Z}^{*b}$ (a więc losowanych z powtórzeniami ze zbioru $\mathbf{Z}$) uczymy model, którego predykcje dla wektora x wynoszą $\hat{f}^{*b}(x)$. Poprzez odpowiednią agregację tak uzyskanych predykcji otrzymamy model *bagging*-owy redukujący wariancję pojedynczych B modeli. Najczęściej używanym agregatem jest średnia:

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x).$$

Dla zadania klasyfikacji K-etykietowej, której ostatecznym wynikiem jest wskazanie na konkretną klasę, wyznacza się estymatory $\hat{f}^k_{bag}(x)$: predykcje przynależności obserwacji x do klasy k, $k = 1, \ldots, K$. Klasyfikator uzyskany poprzez bagging B drzew przyjmuje postać:

$$\hat{G}_{bag}(x) = \arg\max_k \hat{f}^k_{bag}(x).$$

Zauważmy, że jeśli decyzja i-tego klasyfikatora spośród B drzew jest zmienną losową f_i o wariancji σ^2 oraz zmienne losowe $f_1, \ldots, f_B$ są niezależne o tym samym rozkładzie,

[5]Jeśli C jest zmienną losową opisującą liczbę poprawnych decyzji n niezależnych klasyfikatorów, z których każdy podejmuje poprawną decyzję z prawdopodobieństwem $p > 1/2$, to zmienną tą opisuje rozkład $Bin(n, p)$. Wówczas wystarczy zauważyć, że $\mathbb{P}(C > t) \xrightarrow{n\to\infty} 1$

to wówczas:

$$\mathbb{V}ar\left(\frac{1}{B}\sum_{b=1}^{B} f_i\right) = \frac{\sigma^2}{B}$$

Widzimy zatem, że wariancja ostatecznej decyzji, uzyskanej jako średnia ze zmiennych losowych $f_1, \ldots, f_B$, maleje do zera wraz ze wzrostem liczby drzew, o ile drzewa te są niezależne. Zakładając istnienie pomiędzy dowolnymi zmiennymi f_i i f_j $(i \neq j)$ pewnej dodatniej[6] korelacji $\mathbb{C}or(f_i, f_j) = \rho_{ij} = \rho$ (takiej samej dla wszystkich par drzew) otrzymamy:

$$\begin{aligned}
\mathbb{V}ar\left(\frac{1}{B}\sum_{b=1}^{B} f_b\right) &= \frac{1}{B^2}\left(B \cdot \mathbb{V}ar(f_1) + \sum_{i\neq j} \mathbb{C}ov(f_j, f_j)\right) = \\
&= \frac{1}{B^2}\left(B\sigma^2 + \sum_{i=1}^{B}\sum_{j\neq i}^{B} \rho \cdot \sqrt{\mathbb{V}ar(f_i) \cdot \mathbb{V}ar(f_j)}\right) = \\
&= \frac{1}{B^2}\left(B\sigma^2 + B(B-1)\rho\sigma^2\right) = \\
&= \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2
\end{aligned}$$

Wówczas wraz ze wzrostem liczby drzew B drugie wyrażenie w powyższej sumie zanika, a zysk z uśredniania decyzji wszystkich drzew zależy od wielkości współczynnika korelacji poszczególnych drzew i wariancji każdego z nich.

Spostrzeżenie to pozwala poprawić redukcję wariancji baggingu poprzez zminimalizowanie korelacji między drzewami. Celem uzyskania wielu słabo zależnych drzew, we wszystkich węzłach-rodzicach każdego drzewa podział na węzły-dzieci poprzedzony jest losowym wyborem podzbioru zmiennych, które mogą być kandydatami do podziału. Szczegółowy algorytm budowy drzew w modelu lasów losowych, zaprezentowany przez Leo Breimana [2], przedstawiony jest poniżej.

Algorytm 2.2. Budowa lasów losowych

1. wylosuj ze zwracaniem z n-elementowej próby uczącej $\mathbf{Z}$ n wektorów obserwacji $(\mathbf{x}_i, y_i)$ do pseudopróby uczącej $\mathbf{Z}^*$ (na podstawie której zostanie zbudowane drzewo),

2. w każdym węźle budowanego drzewa podział podpróby, która dotarła do tego węzła, przebiega jak niżej:

   (a) niezależnie od innych losowań wylosuj bez zwracania $m \ll p$ predyktorów wektora obserwacji,

[6] Zauważmy, iż z faktu, że wariancja jest nieujemna wynika, że $\rho > -\frac{1}{B-1}$, co przy rosnącej liczbie drzew przybliża się do $\rho > 0$.

(b) zastosuj przyjętą regułę podziału do wylosowanych m predyktorów. Podział opieramy tylko na wylosowanych atrybutach, nie zaś wszystkich p i wśród tych m poszukujemy najlepszego podziału na najlepszym predyktorze (w sensie miary (2.14)).

3. Budujemy drzewo bez przycinania (jeśli to możliwe, do otrzymania liści o elementach pseudopróby uczącej z tylko jednej klasy).

Powyższa procedura opisuje stworzenie jednego modelu drzewa decyzyjnego. Wykonanie tej procedury B-krotnie daje nam B słabo zależnych od siebie modeli o dość dużej wariancji. Końcowa decyzja klasyfikacyjna jest wynikiem „głosowania" wszystkich drzew.

Uwaga 2.2.1. Jedynym parametrem metody, który wymaga ustalenia jest $m \ll p$. Przyjęło się sądzić, że najlepszym wyborem jest (mniej więcej) $m = \sqrt{p}$ (za [8]).

Uwaga 2.2.2. Jako miarę wybierającą najlepszy predyktor i jego punkt odcięcia w problemie (2.15) najczęściej wybierany jest indeks Giniego (taką miarą posługują się m.in. twórcy pakietu `randomForest`).

Uwaga 2.2.3. Lasy losowe są obiecującym rozwiązaniem w problemach z wektorem obserwacji o bardzo dużym rozmiarze (również $p \gg n$), czego uzasadnieniem jest krok drugi powyższego algorytmu.

Out of bag

Zauważmy, że losując ze zwracaniem z n-elementowej próby n obserwacji, średnio ok. 1/3 obserwacji nie jest wylosowana ani razu. Wynika to z następującej dedukcji: prawdopodobieństwo, że i-ty element próby nie zostanie wylosowany wynosi $1 - \frac{1}{n}$. Losując n razy ze zwracaniem po jednym elemencie, prawdopodobieństwo zdarzenia A_i - „i-ty element nie znajduje się w losowanej pseudopróbie" wynosi:

$$\mathbb{P}(A_i) = (1 - \frac{1}{n})^n \xrightarrow{n\to\infty} e^{-1} \approx 0.368.$$

A zatem średnio ok. 37% obserwacji nie znajdzie się w próbie bootstrapowej. Klasyfikując te obserwacje za pomocą drzew, w których tworzeniu nie brały udziału, otrzymamy nieobciążony estymator dokonania błędnej klasyfikacji przez las losowy, znany pod nazwą OOB (ang. *out-of-bag error*).

Próba o niezrównoważonych klasach

Losując ze zwracaniem z próby niezrównoważonej istnieje duże prawdopodobieństwo wystąpienia zaledwie kilku lub nawet żadnego reprezentanta klasy o mniejszej liczebności w próbie bootstrapowej. W rezultacie otrzymamy klasyfikator obciążony w kierunku

klasy dominującej. Częstym obejściem tego problemu jest zrównoważenie klas poprzez zmniejszenie liczby reprezentantów w klasie dominującej (ang. *downsampling*) lub powielenie wystąpień z klasy rzadkiej *oversampling*. Innym podejściem może być surowsze karanie za niepoprawną klasyfikację obserwacji rzadkiej w liściu. Obie metody znajdują zastosowanie w rozwiązaniach zaproponowanych przez autorów pracy [3] (w tym Leo Breimana), czyli BRF (ang. *balanced random forest*) oraz WRF (ang. *weighted random forest*).

BRF - *downsampling* i *oversampling* nie muszą być tak samo efektywne. Badania [4] dowodzą wysokiej skuteczności pierwszego z podejść i pokazują, że *oversampling* jeśli w ogóle przyczynia się do poprawy, to tylko nieznacznej. Jednocześnie minusem *downsamplingu* jest znaczna utrata informacji. Algorytm BRF modyfikuje podejście w lasach losowych, wykorzystującą oba te fakty. W tej metodzie tworzymy - dla każdego drzewa niezależnie - próbę bootstrapową spośród obserwacji z klasy niedoreprezentowanej, a następnie losujemy z powtórzeniami taką samą liczbę obserwacji z klasy dominującej. Na tak skonstruowanych pseudopróbach budujemy każde drzewo wg. tradycyjnego schematu lasów losowych i agregujemy wyniki wszystkich klasyfikatorów.

WRF - obciążenie klasyfikatora lasów losowych w kierunku klasy dominującej można zrównoważyć przypisując klasom odpowiednie wagi. Wagi te wpływają na wynik ostatecznej predykcji w dwóch miejscach algorytmu. Po pierwsze w każdym węźle podczas wyznaczania współczynnika z kryterium Giniego, po drugie: we wszystkich liściach w czasie predykcji odpowiedniej klasy. Wówczas w miejscu tradycyjnej decyzji większością głosów używa się głosów ważonych. Ważony głos jest iloczynem liczby reprezentantów danej klasy oraz przypisanej tej klasie wagi. Ostateczna decyzja klasyfikatora jest agregacją ważonych głosów uzyskanych ze wszystkich drzew. Wartości wag poszczególnych klas są parametrem, który wymaga dopasowania w procesie uczenia.

Eksperymenty przeprowadzone przez autorów pracy [3] nie wykazały szczególnej przewagi któregoś z podejść.

Rozdział 3

Klasyfikacja danych genomicznych

Cel badania

Motywacją do przeprowadzenia niniejszego badania było zweryfikowanie hipotezy o macierzystości poszczególnych nowotworów. Na potrzeby tej pracy stworzono narzędzie, pozwalające na ocenę stopnia macierzystości danej komórki na podstawie zawartego w niej kodu DNA. Jak wiadomo, kod DNA zawiera pełną informację o każdym procesie życiowym danego organizmu, zatem podejmiemy kroki do wyłuskania zeń *sygnatury*, a więc genów, które mogą silniej od innych świadczyć o macierzystości danej komórki.

W rozdziale tym zamieszczono opis i wyniki badania, w tym znalezioną sygnaturę oraz ocenę stopnia macierzystości komórek nowotworowych. Podstawą analizy było zastosowanie statystycznych metod klasyfikacji binarnej opisanych szerzej w rozdziale *Metody statystyczne*. Zanim przejdziemy do prezentacji rezultatów tej pracy, omówimy szerzej tło biologiczne naszych rozważań.

Tło biologiczne

Kluczowym dla zrozumienia wyników tej pracy jest wprowadzenie i krótkie omówienie następujących terminów biologicznych:

- gen oraz ekspresja genu.
- komórka macierzysta i jej różnicowanie,

Różnicowanie komórki macierzystej

Komórką macierzystą jest najbardziej pierwotna komórka, która pozostaje niezróżnicowana. Dwie własności szczególnie charakteryzują każdą komórkę macierzystą: pierwszą jest zdolność do różnicowania, czyli przekształcanie się w komórki innych typów, a w efekcie do wytworzenia poszczególnych organów. Kolejną właściwością jest zdolność

do proliferacji, czyli przedłużania swojego istnienia poprzez powielenie. Najbardziej znanymi komórkami macierzystymi są embrionalne komórki macierzyste: ESC (ang. *embryonic stem cell*). Posiadają one zdolność różnicowania się w komórki wielu typów, dając początek różnym organom tworzącym cały organizm[1]. Somatyczne komórki macierzyste ulokowane są w rozwiniętych tkankach i mogą dać początek co najwyżej kilku różnym typom komórek o podobnych właściwościach (znanym przykładem są komórki macierzyste krwiotwórcze, znajdujące się w szpiku kostnym). Ze względu na to czy komórka macierzysta może dać początek dowolnemu rodzajowi komórek, pewnej części lub tylko jednemu typowi komórki mówimy o komórkach (odpowiednio) pluripotencjalnych, multipotencjalnych lub unipotencjalnych[2].

W swojej pracy wykorzystałam dane uzyskane z indukowanych pluripotencjalnych komórek macierzystych (iPS (ang. *induced pluripotent stem cells*)). Są to pluripotencjalne komórki wytworzone (sztucznie) z niepluripotencjalnych komórek macierzystych (np. z komórek somatycznych). Diagram 3.1 przedstawia poglądowy zarys różnicowania pluripotencjalnej komórki macierzystej.

Istotą tego zjawiska (z perspektywy badań zawartych w niniejszej pracy) jest wyróżnienie różnych stadiów zróżnicowania pluripotencjalnej komórki macierzystej. W dalszej części pracy będziemy posługiwali się następującymi etykietami na nazwanie poszczególnych stadiów (za opisem zawartym na stronie [6])

- SC (ang. *stem cell*) - pluripotencjalna komórka macierzysta, niezróżnicowana.
- EB (ang. *embryoid body*) - ciało embrionalne. Pierwszy, podstawowy etap różnicowania.

Kolejne etapy różnicowania, będące przyczynkiem do formułowania się bardziej zaawansowanych i wyspecyfikowanych struktur organizmu (organów) to:

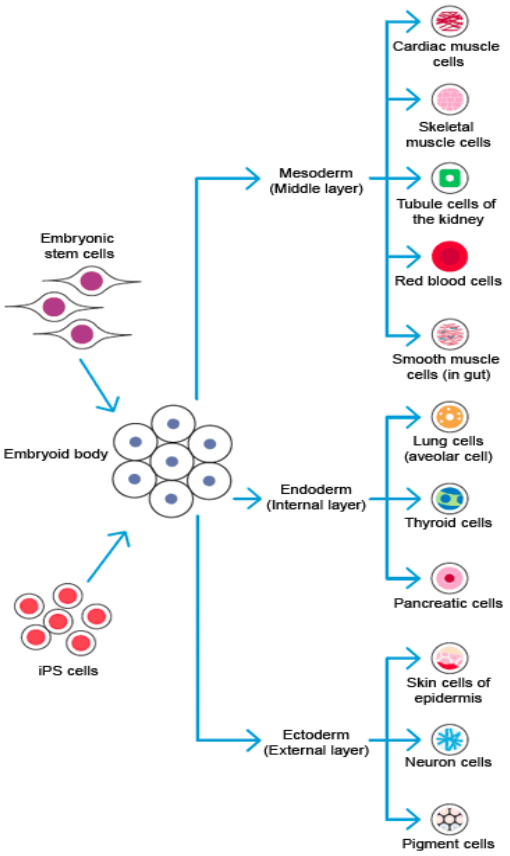
- ECTO (ang. *ectoderm*) - ektoderma, zewnętrzna warstwa zarodkowa,
- MESO (ang. *mesoderm*) - mezoderma, środkowa warstwa komórek zarodka,
- ENDO (ang. *endoderm*)[3] - endoderma, czyli warstwa wewnętrzna.

Więcej szczegółów dotyczących tego zagadnienia można znaleźć w [10].

[1]Przypomnijmy, że zgodnie z najbardziej podstawową systematyką, organizm składa się z wielu organów. Każdy organ składa się z tkanek o określonej postaci i spełniających określoną funkcję, tkanki zaś zbudowane są z komórek i to w niej zachodzą podstawowe funkcje życiowe.

[2]Komórki pluripotencjalne nie mogą różnicować się jedynie w łożysko. Komórki, które różnicują się w dowolny typ, w tym również w łożysko, nazywane są totipotencjalnymi.

[3]Dane, na których bazują dalsze wyniki, zawierają obserwacje uzyskane z późniejszego stadium: DE (ang. *definitive endoderm*).



Rysunek 3.1: *Różnicowanie pluripotencjalnej komórki macierzystej. Zarówno embrionalne komórki macierzyste (ESC) jak i indukowane pluripotencjalne (iPS) różnicują się w ciało embrionalne, a następnie przechodząc w kolejne stadia różnicowania. Odbywają się one równolegle i prowadzą do powstawania tkanek o wyspecyfikowanych zadaniach. I tak różnicowanie środkowej warstwy zarodka: mezodermy prowadzi do powstania komórek mięśnia sercowego, krwi, kanalików nerkowych czy też mięśni. Różnicowanie wewnętrznej warstwy - endodermy - prowadzi do powstania m.in.: tarczycy, płuc i trzustki. Ektoderma jest zewnętrzną strukturą, z której wyspecyfikuje się układ nerwowy, skóra oraz pigment. Diagram pochodzi z broszury na stronie [10]*

Gen oraz ekspresja genu (ang. *gene expression*)

Geny mają ogromne znaczenie w badaniu chorób nowotworowych, jednak ich precyzyjna definicja zależy od specjalizacji w ramach której terminu tego się używa. Na potrzeby tej pracy wystarczy przyjąć, że gen jest pewnym określonym miejscem na nici DNA. Ekspresja genu jest przepisaniem informacji genetycznej na produkt genu, którym są różne formy RNA lub białko. Zależy ona m.in. od rodzaju komórki czy też fazy rozwoju organizmu. W zależności od technologii pozyskiwania danych genetycznych dysponujemy następującymi platformami:

- *methylation* - ekspresja genów regulowana jest metylacją DNA. Wartościami dla danej próbki jest procent zmetylowania danego genu,

- *mRNA* - mechanizm ten mierzy poziom koncentracji genów poprzez sekwencjonowanie mRNA. Wartościami dla danej próbki jest zliczenie wystąpień poszczególnych genów.

Więcej o typach platform oraz o technologii pozyskiwania danych genetycznych przeczytać można w pozycji [7]. W niniejszej pracy ograniczymy się do stwierdzenia, że **gen** jest zmienną losową (o rozkładzie wynikającym z techniki pozyskiwania danych) zaś **ekspresja genu** jest realizacją tej zmiennej losowej. Przez **kod DNA** rozumiemy wektor zmiennych losowych, zaś **profilem DNA** nazwiemy wektor realizacji zmiennych losowych. **Sygnaturą genetyczną** będzie poszukiwany podzbiór najlepszych - w sensie predykcyjnym - genów. Charakterystyczną cechą danych genetycznych, zasługującą na podkreślenie, jest ogromna liczba zmiennych (genów) przy stosunkowo niewielkiej liczbie pobranych próbek - obserwacji.

Przygotowanie danych

Źródło i opis danych

Klasyfikator testowany będzie na pacjentach dotkniętych różnymi chorobami nowotworowymi, celem określenia profilu macierzystości poszczególnych nowotworów. W tym celu posłużono się profilami DNA tkanek nowotworowych (TCGA) oraz indukowanych pluripotencjalnych komórek w różnym stadium zróżnicowania (PCBC).

PCBC (ang. *Progenitor Cell Biology Consortium*): zbiór zawierający dane o profilach DNA komórek o różnych stopniach zróżnicowania [6]. Zarówno dane jak i metadane, dostarczające informacji o stopniu zróżnicowania danej komórki, znaleźć można na stronie `www.synapse.org/pcbc`.

TCGA (ang. *The Cancer Genome Atlas*): zbiór zawiera obszerne informacje o pacjentach dotkniętych chorobami nowotworowymi [16]. W szczególności zbiory te zawierają dane o ekspresji genów w komórkach zmutowanych tkanek a także (kontrolne) ekspresje genów w tkankach zdrowych. Na potrzeby tej pracy skorzystano z danych z pakietu RTCGA [13]. Dodatkowo obserwacje pochodzące z nowotworów poszczególnych organów (takich jak: glejak, nowotwór jajnika, nerki, etc.) zostały zebrane w trzy grupy: mezodermy, ektodermy i endodermy, ze względu na typ zaatakowanej tkanki. Pełny spis nowotworów należących do poszczególnych grup wraz z objaśnionymi symbolami, zawarty jest w dodatku na końcu tej pracy.

Zarówno PCBC jak i TCGA dostarczają informacje o ekspresji danych na platformach: *methylation* i *mRNA*.

Charakterystyka danych

Na podstawie metadanych o zbiorach PCBC z platform *methylation* oraz *mRNA* otrzymujemy informację o dostępnych stadiach zróżnicowania komórek, a także o liczbie obserwacji pochodzących z danego stadium. Podobne zestawienie, jednak uwzględniające typ nowotworu, możemy uczynić dla danych TCGA dostępnych z pakietem `RTCGA`.

PCBC	SC	EB	ECTO	MESO	DE	Σ
methylation	44	22	11	11	11	99
mRNA	78	49	29	40	33	229

(a) *Dane* PCBC

TCGA	ENDO		MESO		Σ
	tumor	normal	tumor	normal	
methylation	540	118	1425	241	2822
mRNA	408	22	1189	60	1733

(b) *Dane z pakietu* `RTCGA`

Tabela 3.1: *Liczba i typy obserwacji w danych*

Podział na kolumny: ENDO i MESO wynika z tego z jakiej warstwy zarodkowej wykształcił się zaatakowany danym nowotworem organ i nie należy go utożsamiać ze zdefiniowanymi wcześniej etapami różnicowania komórek macierzystych. Szczegółowy wykaz nowotworów z ustaleniem odpowiadającej mu warstwy zarodkowej zamieszczony został w dodatku na końcu tej pracy.

Wprowadźmy następujące oznaczenia dla poszczególnych typów obserwacji:

PCBC(`G`) - podzbiór wszystkich obserwacji zbioru PCBC, zawierający obserwacje o stopniu zróżnicowania `G`, gdzie `G` ∈ `{SC, EB, MESO, ECTO, DE}`,

TCGA(`Z, tumor`) - podzbiór wszystkich obserwacji zbioru TCGA, zawierający komórki pobrane z organu powstałego z zarodkowej warstwy typu `Z`, gdzie `Z` ∈ `{ENDO, MESO}` i dotkniętego nowotworem.

TCGA(`Z, normal`) - analogicznie zdefiniowany podzbiór obserwacji określający komórki kontrolne niedotknięte nowotworem.

Należy zaznaczyć, że zarówno dane TCGA jak i PCBC charakteryzuje duża liczba predyktorów: znacznie większa od liczby obserwacji. Pełną informację o liczbie obserwowanych zmiennych na poszczególnych platformach przedstawia poniższa tabelka.

	PCBC	TCGA	PCBC ∩ TCGA
methylation	23384	27580	23381
mRNA	12948	17816	10640

Tabela 3.2: *Liczba obserwowanych zmiennych w zbiorach PCBC i TCGA oraz liczba zmiennych wspólnych na obu zbiorach*

Normalizacja oraz kompozycja zbiorów danych

Z zamiarem konkatenacji zbiorów pochodzących z różnych eksperymentów wiąże się duży problem dotyczący zastosowanych do nich metod normalizacyjnych danych. Z tego powodu dane PCBC i TCGA zostały sprowadzone do standaryzowanych rang (statystyk pozycyjnych) danej wartości w obrębie genu. Następnym krokiem było przypisanie obserwacjom odpowiedniej klasy. Celem nauczenia modelu weryfikującego macierzysty charakter danej próbki, klasę *sukces* - rozumiany jako zdarzenie w próbie Bernoulliego - przypisujemy obserwacjom PCBC(`SC`). Klasę tę oznaczymy etykietą SC. Pozostałym obserwacjom z tego zbioru, a więc PCBC(∼`SC`), przypisujemy klasę *porażka*. Zbiór ten powiększymy o obserwacje TCGA(`normal`), które - pochodząc z komórek zdrowych organów - są komórkami zróżnicowanymi.

Typ zdarzenia	Oznaczenie	Reprezentanci danej klasy
sukces	*SC*	PCBC(`SC`)
porażka	*nonSC*	PCBC(∼`SC`) + TCGA(`normal`)

Tabela 3.3: *Określenie typu zdarzeń binarnych oraz definicja ich reprezentantów*

Schemat 3.2 ilustruje proces przygotowania i kompozycji danych zbiorów.

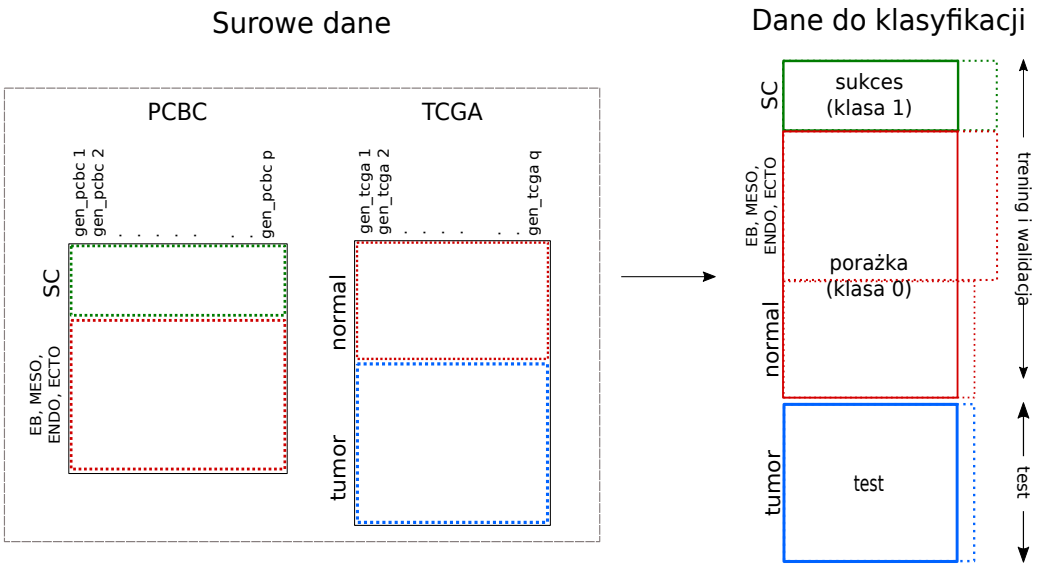
Wyniki

Na 4 różnych zbiorach danych {*methylation*, *mRNA*} × {TCGA(`ENDO, normal`), TCGA(`MESO, normal`)} porównano ze sobą 2 metody klasyfikacji binarnej: regresję logistyczną z regularyzacją LASSO oraz lasy losowe a także przeprowadzono procedurę wyboru sygnatury genetycznej. Do wygenerowania wszystkich wyników zawartych w tej pracy posłużył pakiet `RStemnessScorer`. Odpowiedni kod programu w języku R zawiera Dodatek B.

Budowa modeli

Szkic postępowania

Skonkatenowany zbiór został unormowany w następujący sposób: każdej obserwacji w obrębie danej zmiennej przypisano jej rangę podzieloną przez liczbę obserwacji. Tak

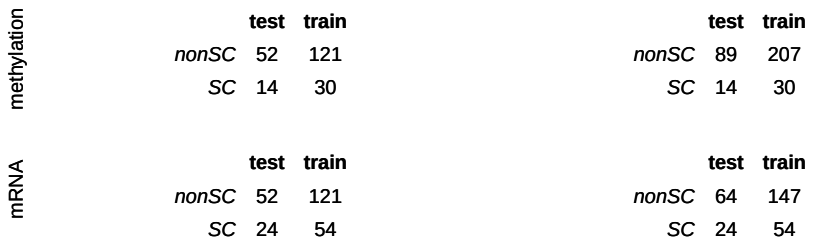


Rysunek 3.2: *Przygotowanie danych. Po znormalizowaniu poprzez rangi oraz wybraniu genów wspólnych na obu zbiorach, następuje wyodrębnienie części obserwacji* TCGA*(`tumor`), na której chcemy testować klasyfikator. Do nauczenia klasyfikatora tworzymy zbiór treningowy i walidacyjny. Wykorzystamy do tego celu dane o macierzystości* PCBC*. Dodatkowo poszerzymy ten zbiór o kontrolne obserwacje* TCGA*(`normal`). Zmienną odpowiedzi będzie wektor binarny o długości odpowiadającej liczbie obserwacji, posiadający wartość równą* 1*, gdy obserwacja jest klasy SC i* 0 *w przeciwnym przypadku.*

unormowany zbiór podzielony został na część treningową (70% obserwacji) i walidacyjną (30% obserwacji). Następnie zbudowano modele **regresji logistycznej z karą LASSO**. Parametr kary λ używany w metodzie LASSO wyznaczony został kroswalidacyjnie przy trzech rodzajach funkcji celu: minimalizacja dewiancji, maksymalizacja AUC oraz maksymalizacja frakcji poprawnej klasyfikacji. Pozwalało to na wyznaczenie i porównanie trzech różnych sygnatur uzyskanej z jednej metody regresji logistycznej. Kolejnymi nauczonymi modelami były **lasy losowe** oraz **lasy losowe z oversamplingiem** czyli z dolosowywaniem obserwacji z niedoreprezentowanej klasy, aby zbilansować rozkład sukcesów i porażek w zbiorze treningowym. Interesującą nas sygnaturę otrzymaliśmy na podstawie wskaźnika Giniego. Ostatnim krokiem była analiza jakościowa każdego z klasyfikatorów. Do oceny jakości użyto krzywej ROC oraz dewiancji. W związku z przypuszczeniem o przeuczeniu modeli przeprowadzono również y-randomizację.

Zbiór treningowy i walidacyjny

Podział zbioru uczącego na treningowy (`train`) i walidacyjny (`valid`) wykonany został w stosunku 7:3 odpowiednio, z proporcjonalnym zachowaniem udziału klasy zaintere-



Rysunek 3.3: *Liczba reprezentantów klasy sukcesu SC i porażki nonSC w zbiorze treningowym*

sowania SC. Zestawienie na rysunku 3.3 pokazuje szczegółowo liczebność obserwacji.

Oznaczmy dodatkowo macierz obserwacji dla poszczególnych genów (w wierszach obserwacje, w kolumnach geny) poprzez `train$x` oraz wektor odpowiedzi jako `train$y` dla zbioru treningowego i dla zbioru walidacyjnego w odpowiedni sposób. Wszystkie modele zostały nauczone na zbiorze treningowym `train`.

Model regresji logistycznej

Współczynniki w modelu regresji logistycznej z karą LASSO zostały ustalone przy trzech różnych parametrach regularyzacyjnych $\lambda_1, \lambda_2, \lambda_3$. Odpowiednie parametry λ_i ustalono w procedurze kroswalidacyjnej przy optymalizacji następujących funkcji celu:

a) dewiancja (ozn. *Deviance*) - minimalizowana,

b) AUC - maksymalizowana,

c) misklasyfikacja (ozn. *Class*) - minimalizowana.

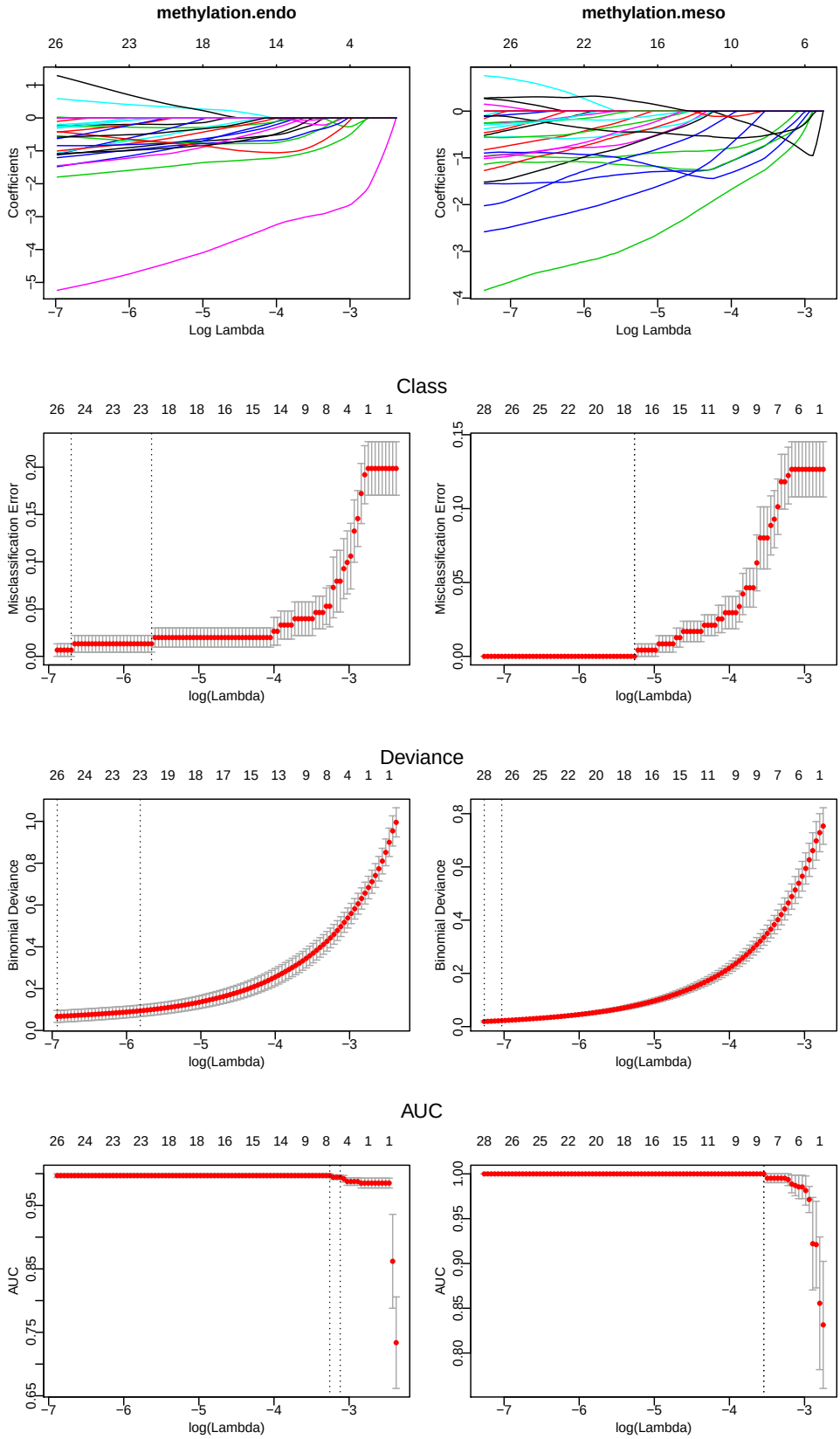
Do tego celu użyto funkcji `cv.glmnet{glmnet}`. Rysunki 3.4 oraz 3.5 przedstawiają zarówno wyznaczone ścieżki współczynników jak i wartości funkcji celu dla poszczególnych parametrów regularyzacyjnych λ_i.

Spośród wszystkich możliwych współczynników wybieramy te, dla których odpowiednia funkcja celu osiąga ekstremum.

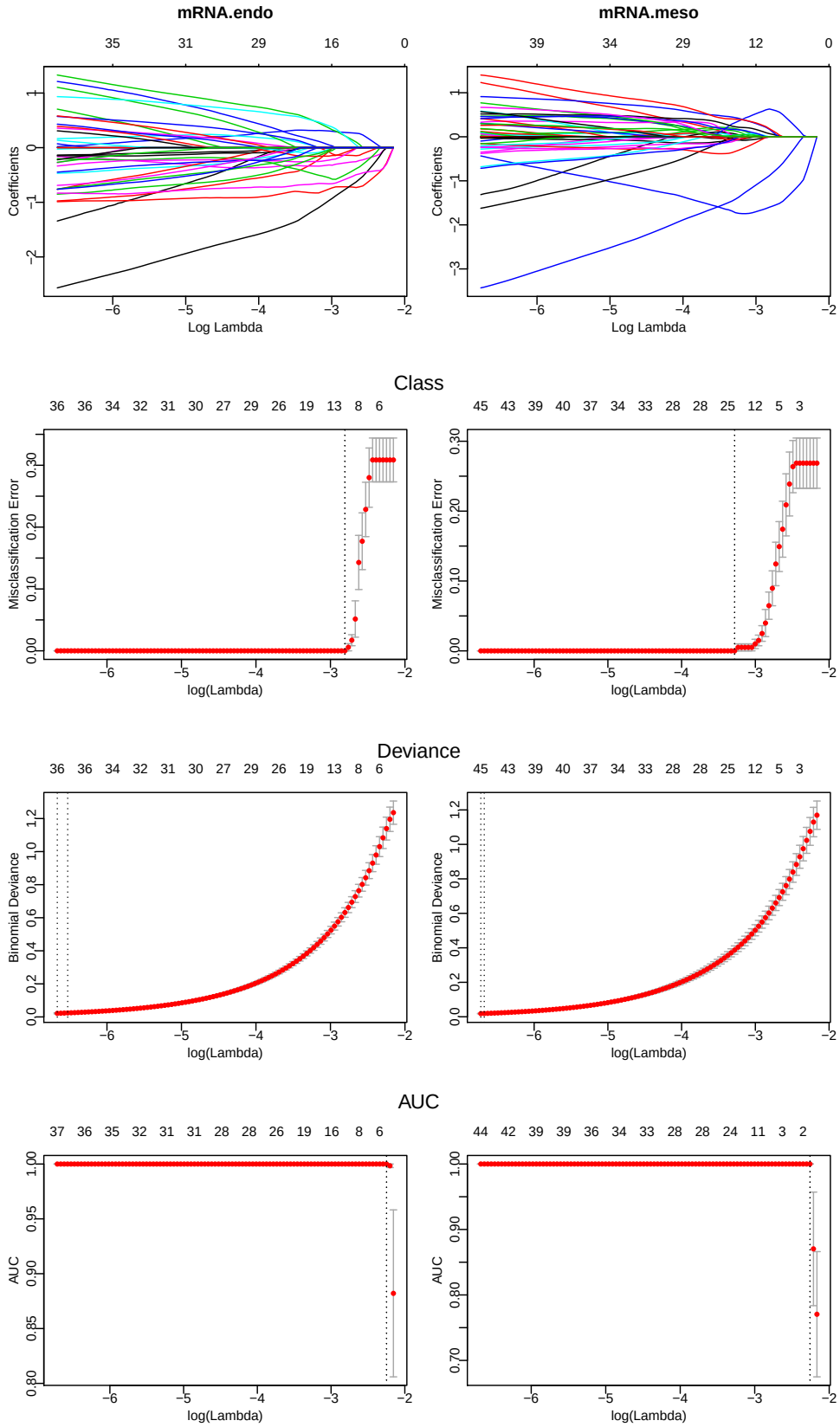
Modele lasów losowych

W tym miejscu zdecydowano się na porównanie wyników z trzech modeli: pierwszy z nich zbudowany został na oryginalnych obserwacjach ze zbioru treningowego, tzn. zachowując niezbilansowany rozkład klas (ozn. RF), drugi na danych o zrównoważonym - poprzez powielenie obserwacji z klasy niedoreprezentowanej - rozkładzie klas (ozn. $RF + oversampling$), a trzeci jestem modelem zbilansowanym przez algorytm BRF (ozn. BRF).

Na każdy z modeli składało się 5000 drzew. Liczba genów losowanych do każdego węzła w każdym drzewie była pierwiastkiem kwadratowym liczby predyktorów. Rysunek

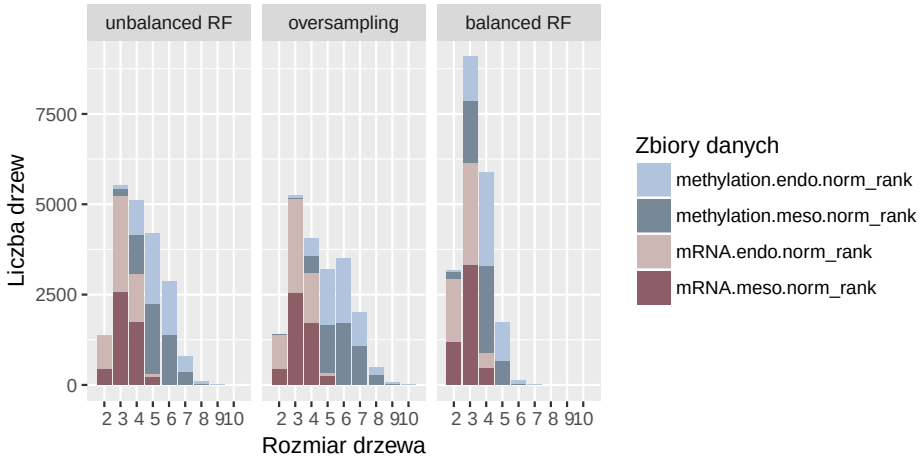


Rysunek 3.4: *Ścieżki współczynników oraz wartości poszczególnych funkcji celu dla różnych parametrów λ regularyzacji LASSO. Zbiory z platformy methylation*



Rysunek 3.5: *Ścieżki współczynników oraz wartości poszczególnych funkcji celu dla różnych parametrów λ regularyzacji LASSO. Zbiory z platformy mRNA*

3.6 przedstawia rozmiar poszczególnych drzew, liczony jako liczba węzłów decyzyjnych.



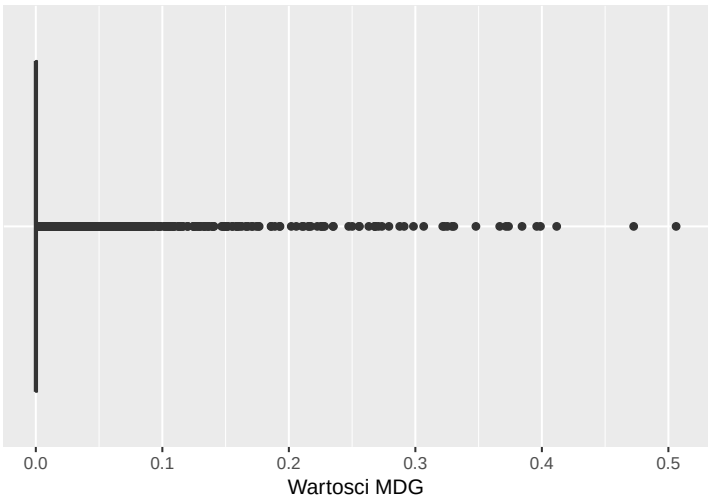
Rysunek 3.6: *Udział liczby drzew o danym rozmiarze w poszczególnych klasyfikatorach. Do budowy każdego z lasów ustalono liczbę 5000 drzew. Kolorami wyróżniono różne zbiory treningowe. Po lewej stronie widzimy wyniki dla modelu zrobionego na oryginalnych, niezbilansowanych danych, na środku dla modelu z oversamplingiem, zaś po prawej z wykorzystaniem procedury BRF*

Celem zweryfikowania, które ze zmiennych model uważa za istotne, przyjrzeliśmy się wykresom wartości $MDG(X_i)$ (ang. *Mean Decrease Gini*) dla każdej cechy X_i przy czym $i = 1, 2, \ldots, q$ gdzie q - liczba wszystkich kolumn w danym zbiorze treningowym. Czym wyższa wartość tego współczynnika w danej cesze, tym istotniejszy jej wpływ w klasyfikacji. Przykładowy rozkład wszystkich wartości $MDG(X_i)$ wyliczony dla zbiorów `methylation_meso` przedstawiony jest na rysunku 3.7a. Pokazuje on, iż w istocie jest kilka zmiennych o wysokiej wartości tego współczynnika, jednak ok. 75% obserwacji jest równych, lub prawie równych zero. Wykres 3.7b ilustruje odsetek zmiennych, których wartość MDG wynosi dokładnie zero.

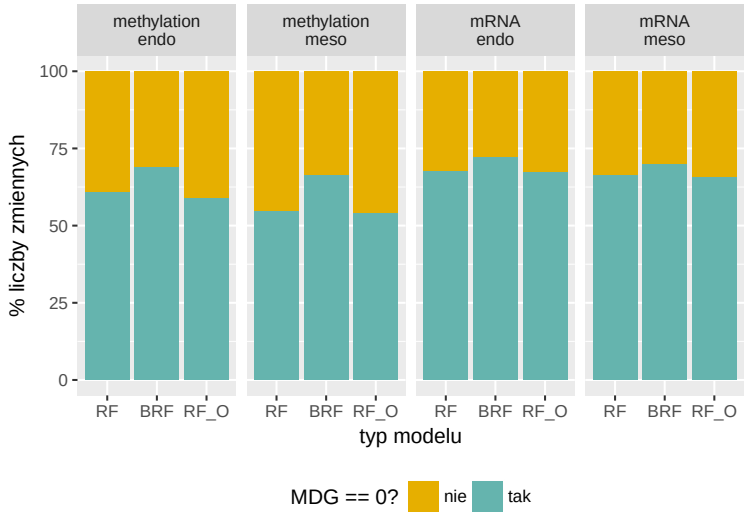
Liczba cech z wartością MDG równą dokładnie 0 jest nad wyraz duża: na 10–20 tysięcy zmiennych, jest to ponad połowa. Nasuwa się zatem pytanie, czy wśród tylu zmiennych mogły być takie, które ani razu nie zostały wylosowane do żadnego węzła w żadnym drzewie? Z bardzo dużym prawdopodobieństwem nie. Otóż, niech p_i $i = 1, 2, \ldots, q$ (q - liczba wszystkich zmiennych) oznacza prawdopodobieństwo, że i-ta zmienna zostanie wylosowana jako jedna z $\sqrt{q}$ zmiennych do danego węzła w drzewie. Wówczas:

$$p_i = \frac{\binom{q}{\sqrt{q}-1}}{\binom{q}{\sqrt{q}}} = \frac{\sqrt{q}}{q - \sqrt{q} + 1} = \frac{1}{\sqrt{q} + \frac{1}{\sqrt{q}} - 1}.$$

Niech N oznacza liczbę wszystkich węzłów, we wszystkich m drzewach, tzn. $N = \sum_{j=1}^{m} t_j$, gdzie t_j liczba węzłów w j-tym drzewie. Prawdopodobieństwo, że i-ta zmienna nie zostanie wylosowana do żadnego węzła, w żadnym m drzew wynosi:



(a) *Przykładowy rozkład wartości MDG dla wszystkich cech w zbiorze methylation.meso*



(b) *Procent liczby cech z wartością MDG dokładnie równą 0 dla trzech rodzajów próbkowania. RF jest tradycyjnym modelem lasów losowych, BRF zbilansowanym poprzez procedurę BRF, a RF_O modelem lasów losowych z oversamplingiem*

Rysunek 3.7: *Miara różnorodności Giniego w drzewach losowych*

$$(1-p_i)^N = \left(1 - \frac{1}{\sqrt{q} + \frac{1}{\sqrt{q}} - 1}\right)^N.$$

Przy liczbie drzew m równej 5000, liczbie zmiennych q wynoszącej (w przybliżeniu) 20000 oraz słabym założeniu, że każde drzewo składa się z jednego węzła, tzn. $N = 5000$ otrzymamy prawdopodobieństwo niewylosowania i-tej zmiennej równe około $3.0 \cdot 10^{-16}$. Jak widzimy, prawdopodobieństwo, że dana zmienna, wśród nawet 20000 innych nie

zostanie wylosowana ani razu jest bardzo małe.

Można więc zatem przypuszczać, że zmienne te, występujące w towarzystwie pozostałych, nie niosą do modelu istotnej informacji.

Sygnatura

Rysunki: 3.8 i 3.9 przedstawiają sygnaturę wybraną z poszczególnych modeli. Dla lasów losowych ustalono model BRF, a kolejność wyboru zadawana jest wielkością MDG (malejąco) dla pierwszych pięćdziesięciu najistotniejszych zmiennych. Wartości współczynników z modeli regresji logistycznej przedstawione są na wykresach słupkowych. Poszczególne kolory oznaczają współczynniki wyznaczone przy różnych parametrach regularyzacyjnych metody LASSO.

Analiza jakości modeli

Jakość stworzonych modeli badano na zbiorze walidacyjnym `valid` poprzez wyznaczenie predykcji na podstawie macierzy `valid$x` i porównanie ich wartości z prawdziwymi (znanymi) etykietami `valid$y`. Przyjrzano się rozkładowi uzyskanych predykcji (Rys. 3.10, 3.11) oraz krzywym ROC (Rys. 3.12). Rezultaty przedstawione na 3.12, choć bardzo pożądane, to jednak sugerują rozważenie hipotezy o przeuczeniu modelu.

Y-permutacja

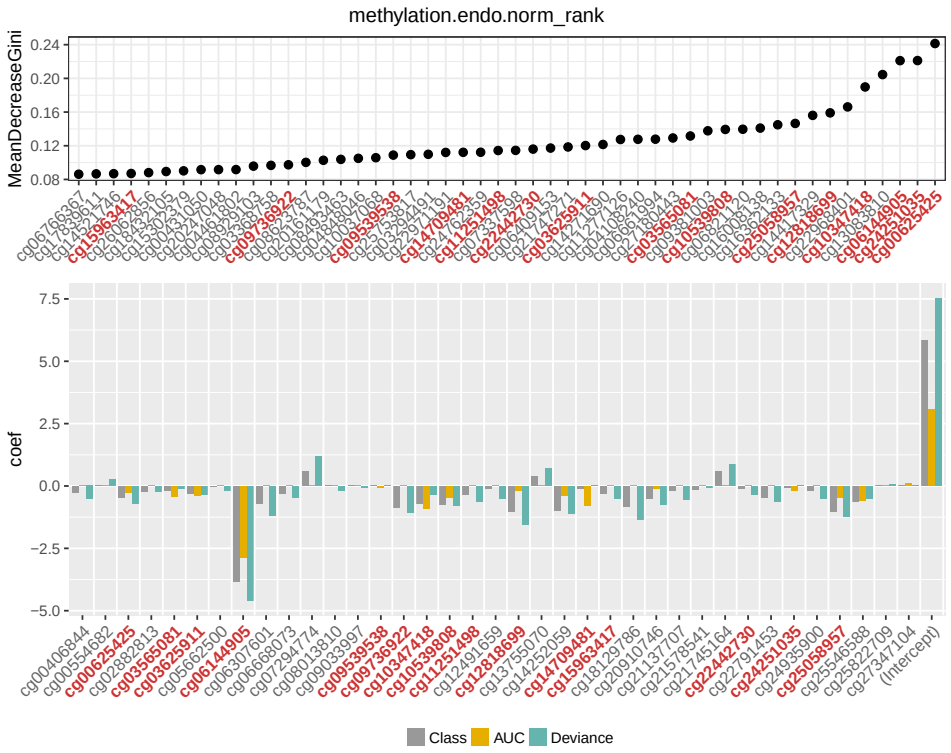
Celem zweryfikowania możliwości predykcyjnej naszych modeli przeprowadzono eksperyment polegający na budowaniu modeli losowych, tzn. poszczególnym obserwacjom `train$x` przypisywano losowo etykiety. Losowość uzyskiwano poprzez permutację elementów wektora odpowiedzi w zbiorze uczącym (`train$y`). Eksperyment z permutowaniem etykiet przeprowadzono 100 razy i obserwowano jak zmieniają się AUC oraz dewiancja uzyskanego (losowego) modelu, wyliczone na danych walidacyjnych `valid` przy jednoczesnym monitorowaniu oraz jak bardzo oryginalny binarny wektor *różni się* od jego przetasowanej wersji. Jedna (`i`-ta) iteracja eksperymentu wyglądała następująco:

i. losowa permutacja elementów wektora odpowiedzi `train$y` (`rand_y <- sample(train$y)`),

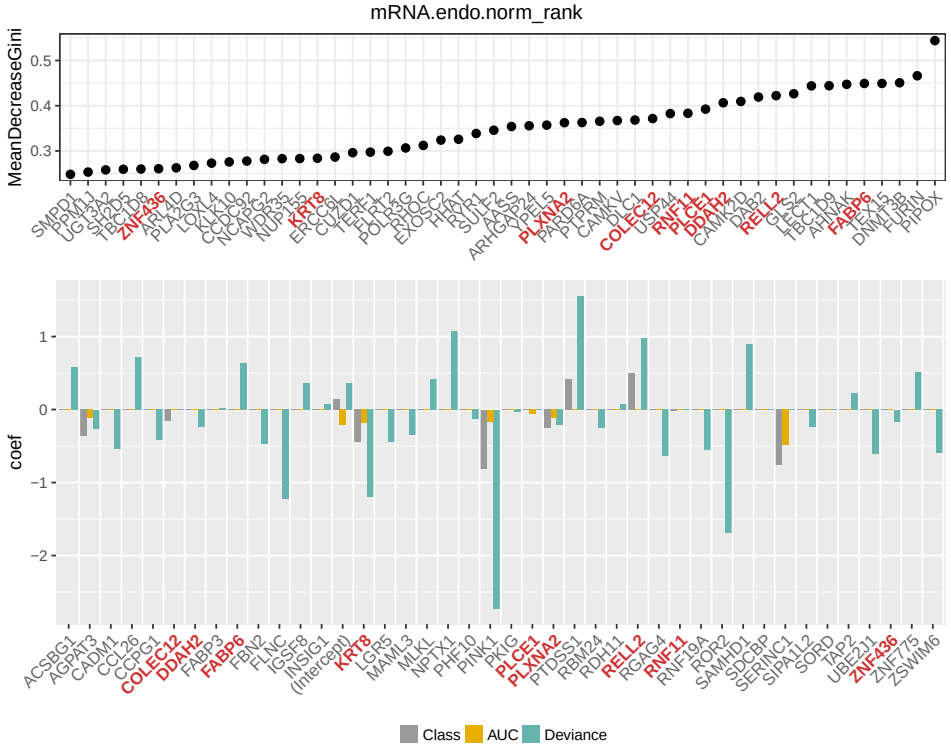
ii. uczenie modeli; dodatkowo uczono model lasów losowych z dolosowywaniem elementów z klasy niedoreprezentowanej (SC), (ang. *oversampling*),

iii. wyznaczenie wektora `p` predykcji dla obserwacji `valid$x`,

iv. obliczenie AUC oraz dewiancji dla wektora `p` i etykiet `valid$y`

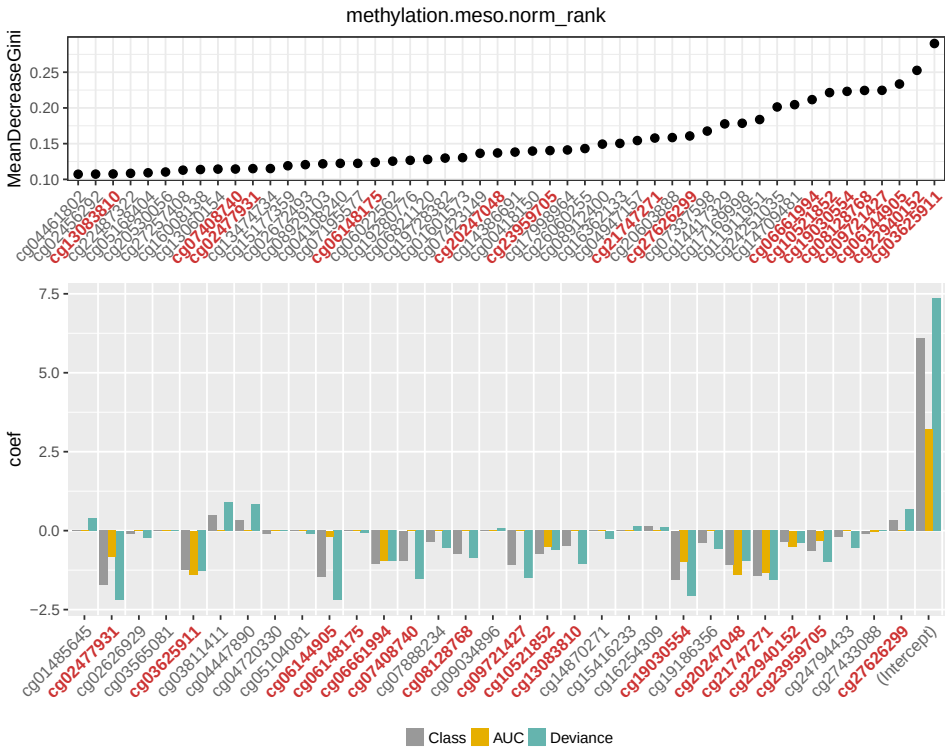


(a) *Platforma* methylation

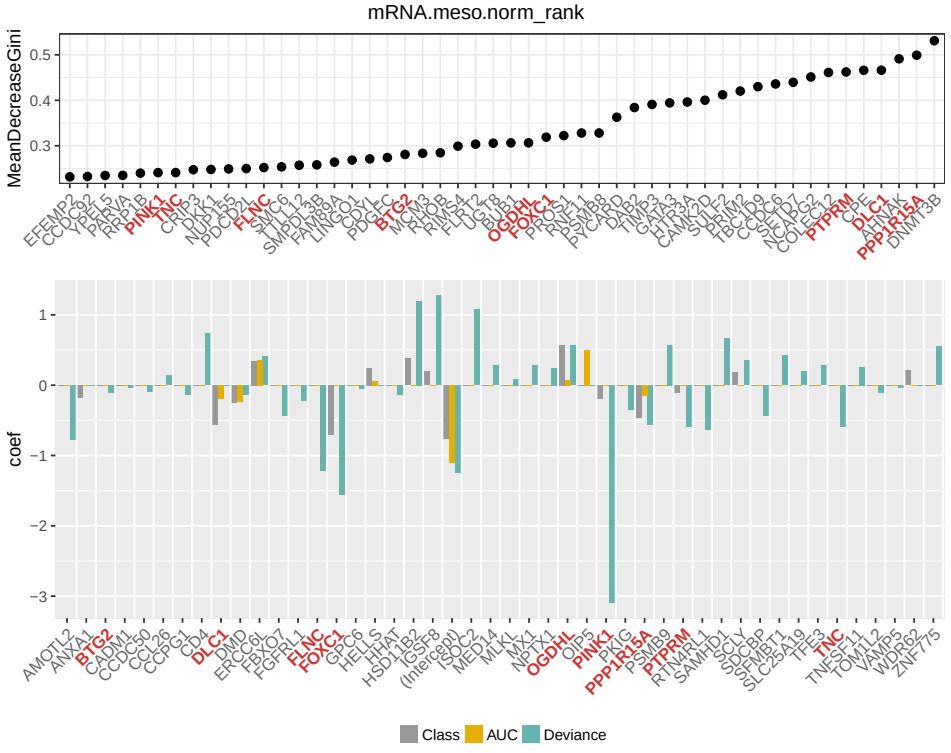


(b) *Platforma* mRNA

Rysunek 3.8: *Sygnatura wyznaczona dla nowotworów z grupy ENDO. Na górze widzimy wykres wartości MDG dla 50 najistotniejszych cech z modelu zbilansowanych lasów losowych (BRF). Dolny rysunek przedstawia wartości wyestymowanych współczynników z modelu regresji logistycznej przy 3 różnych regularyzacjach LASSO. Cechy wspólne dla modelu lasów losowych i regresji logistycznej z regularyzacją, oznaczono kolorem czerwonym*

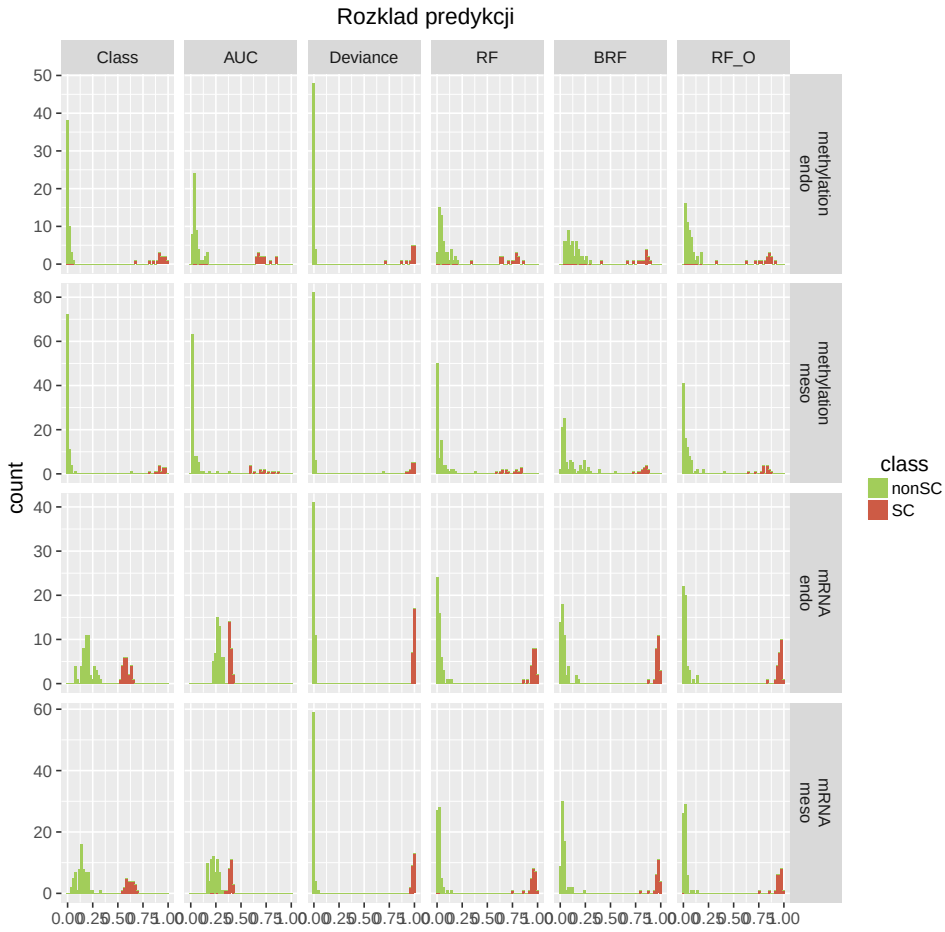


(a) *Platforma* methylation

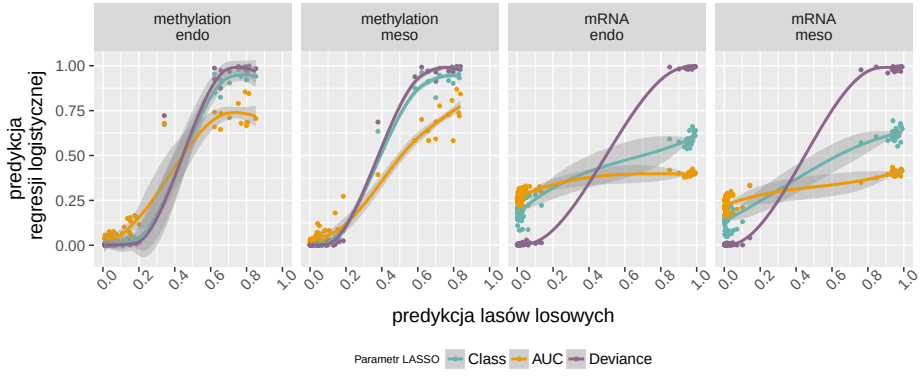


(b) *Platforma* mRNA

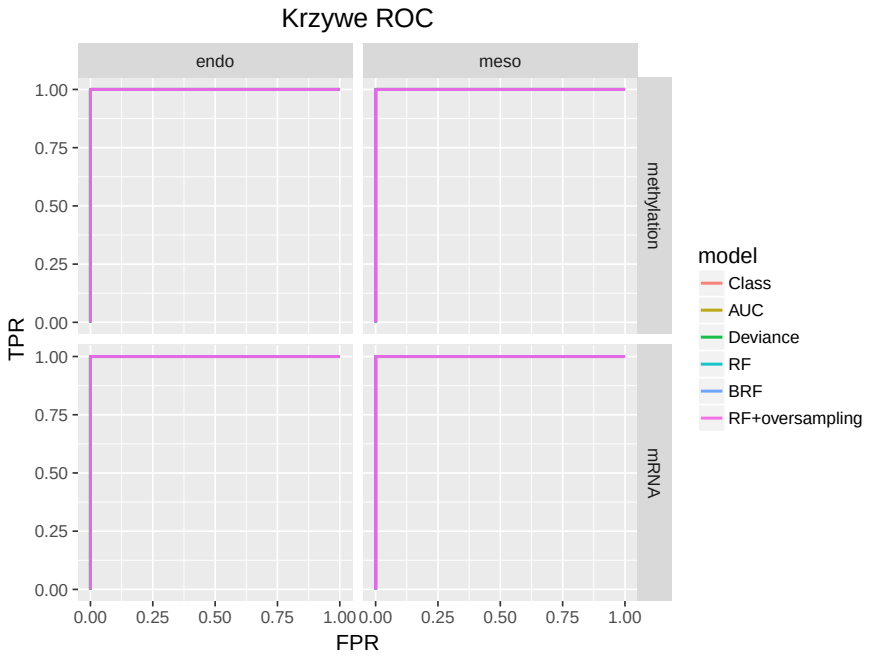
Rysunek 3.9: *Sygnatura wyznaczona dla nowotworów z grupy MESO. Na górze widzimy wykres wartości MDG dla 50 najistotniejszych cech z modelu zbilansowanych lasów losowych (BRF). Dolny rysunek przedstawia wartości wyestymowanych współczynników z modelu regresji logistycznej przy 3 różnych regularyzacjach LASSO. Cechy wspólne dla modelu lasów losowych i regresji logistycznej z regularyzacją, oznaczono kolorem czerwonym*



Rysunek 3.10: *Histogramy z uzyskanych predykcji dla poszczególnych metod na każdym z podzbiorów. Kolorami wyróżniono klasę danej obserwacji. W trzech pierwszych kolumnach widnieją rozkłady uzyskane z modeli regresji logistycznej dla różnych parametrów regularyzacyjnych LASSO, a w trzech kolejnych: predykcje modeli lasów losowych: zwykłych (RF), zbilansowanych (BRF) oraz z oversamplingiem*



Rysunek 3.11: *Rozkład predykcji - zależność predykcji z modeli liniowych do wyników uzyskanych klasyfikatorem lasów losowych (RF) na każdym z 4 podzbiorów*



Rysunek 3.12: *Krzywe ROC według poszczególnych zbiorów danych dla wszystkich klasyfikatorów. Pierwsze 3 są klasyfikatorami regresji logistycznej z LASSO dla 3 różnych parametrów regularyzacyjnych, 3 kolejne są różnymi wariantami lasów losowych*

v. oraz wyznaczenie *niepodobieństwa* dwóch wektorów binarnych: `train$y` i jego permutacji `rand_y`.

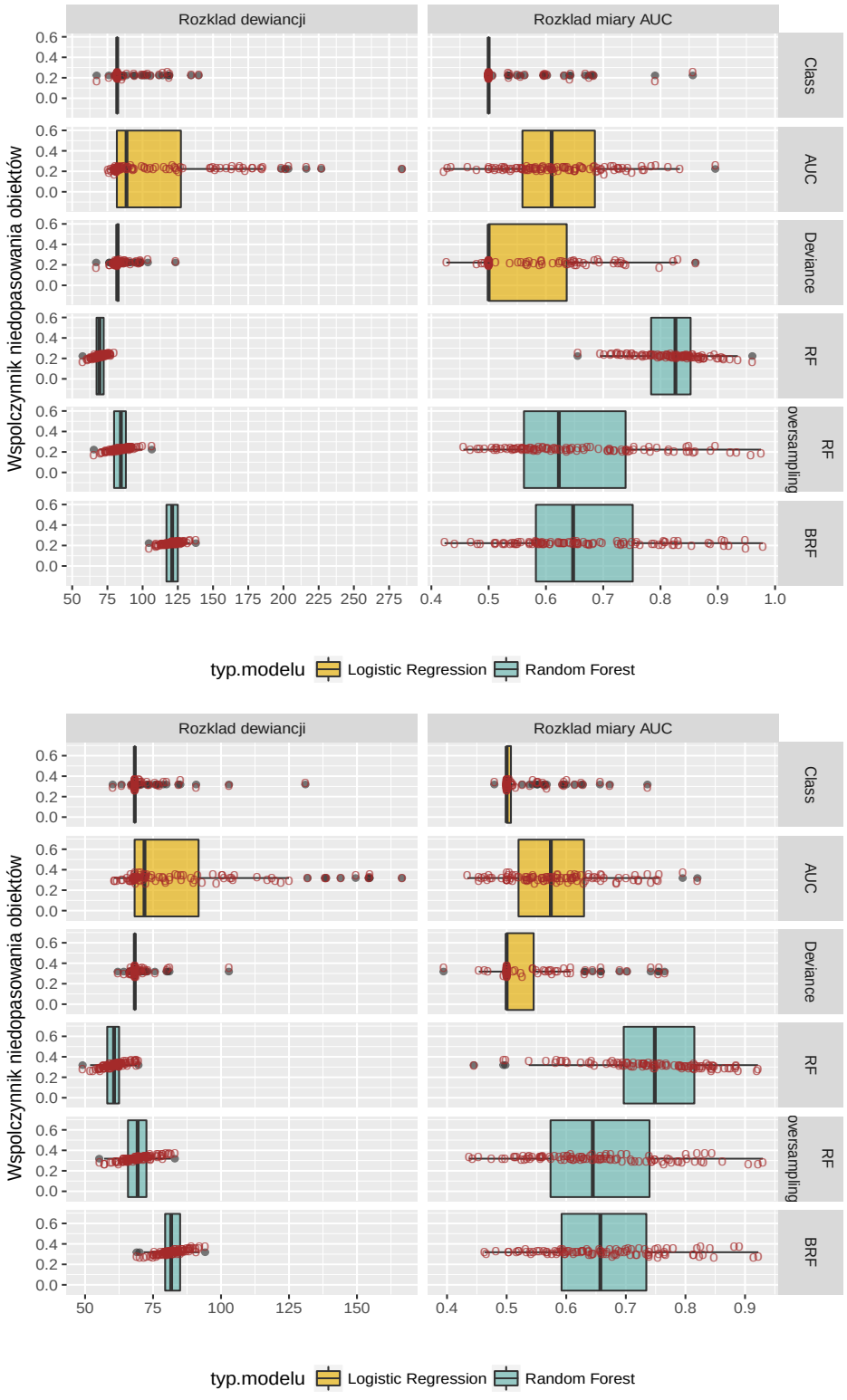
Jako miary *niepodobieństwa* użyto unormowanej odległości Hamminga (zwanej także współczynnikiem dopasowania obiektów [12]), zdefiniowanej jako:

$$H(y^A, y^B) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y_i^A \neq y_i^B)$$

gdzie y^A oraz y^B są dwoma wektorami binarnymi tej samej długości n. Spodziewanym wynikiem nauczenia modelu na danych losowych jest AUC ≈ 0.5 i wysoka dewiancja. Wykresy na Rys. 3.13 przedstawiają rozkłady AUC oraz dewiancji uzyskanych po stukrotnym wykonaniu powyższej procedury.

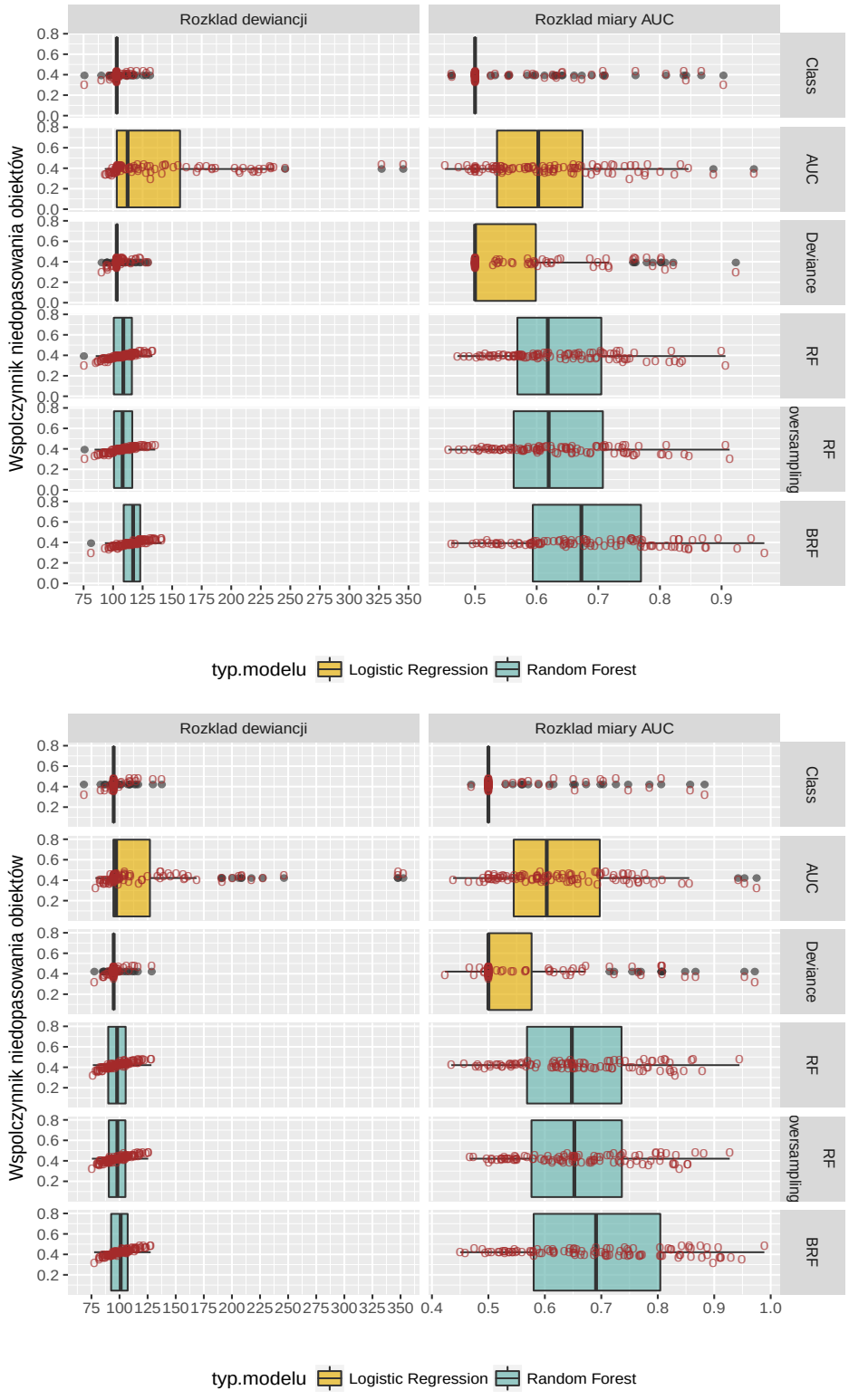
Klasyfikacja komórek nowotworowych

Po analizie jakościowej stworzonych narzędzi, zostały one użyte do oszacowania prawdopodobieństwa macierzystości poszczególnych nowotworów. Posłużono się obserwacjami TCGA(`., tumor`) dla odpowiednich grup ENDO i MESO oraz z obu platform: *methylation* i *mRNA*. Uzyskane wyniki przedstawiono za pomocą wykresów pudełkowych (Rys. 3.14).



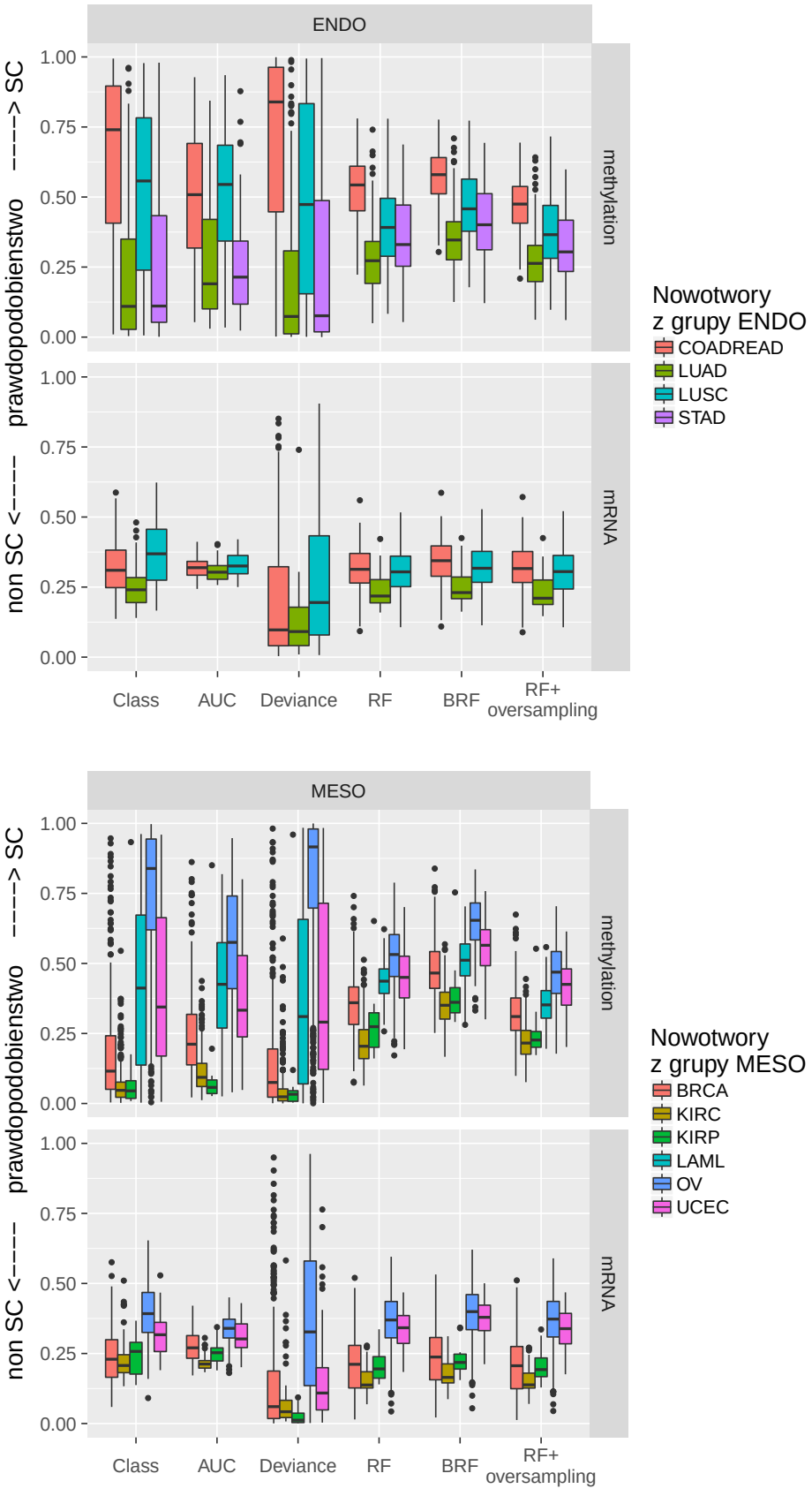
(a) *Platforma* methylation*: grupa MESO na górze i ENDO na dole wykresu*

Rysunek 3.13: *Rozkłady dewiancji i AUC policzone na zbiorze* `valid` *dla 100 różnych potasowań wektora* `train$y` *dla grup nowotworów* MESO *(góra),* ENDO *(dół). Wykresy pudełkowe pokazują położenie pierwszego i trzeciego kwartyla oraz mediany.Symbolem „o" oznaczono wartości unormowanej...*



(b) *Platforma* mRNA*: grupa MESO na górze i ENDO na dole wykresu*

Rysunek 3.13: *...odległości Hamminga dla danego punktu dewiancji/AUC odpowiednio. Prezentowane rysunki pokazują zachowanie regresji logistycznej z regularyzacją L1 dla trzech różnych parametrów* λ *(wykresy żółte) oraz modeli lasów losowych dla różnych prób bootstrapowych (wykresy niebieskie) (kontynuacja)*



Rysunek 3.14: *Wyniki na obserwacjach nowotworowych*

Rozdział 4

Dyskusja

W niniejszej pracy zaprezentowano wyniki uzyskane z dwóch metod klasyfikacji: regresji logistycznej z regularyzacją LASSO (ozn. LR) oraz drzew losowych (ozn. RF), na przykładzie danych genomicznych. Skoncentrowano się na problemie selekcji istotnych zmiennych (sygnatury) przez każdą z metod, przyjrzano się również innym charakterystycznym cechom każdego z algorytmów.

Główne problemy, z którymi się zmierzyliśmy się to:

1. proweniencja danych: PCBC i TCGA - klasyfikator uczony na danych pochodzących z dwóch różnych eksperymentów mógłby wykazywać wrażliwość ze względu na metodę pozyskania danych i stłumić analizowany przez nas efekt,

2. duża utrata informacji poprzez normalizację rangami (problem implikowany przez wyżej postawiony aspekt),

3. niezbilansowane dane,

4. możliwość przeuczenia modelu.

Do klasyfikacji wyróżniono 4 zbiory ze względu na platformę danych (*methylation* i *mRNA*) oraz typ tkanek (ENDO i MESO).

Uwagi ogólne

Pomimo redukcji całej informacji jaką niosły dane PCBC i TCGA do informacji o monotoniczności, wyniki klasyfikacji metodami zarówno LR jak i RF przerosły najśmielsze oczekiwania. Na każdym z 4 wyróżnionych podzbiorów skuteczność klasyfikacji, mierzona poprzez AUC, sięgała 100% lub prawie 100% dla danych MESO-*mRNA* (zgodnie z rys. 3.12). Rysunek 3.10 pokazuje, że nie tylko uzyskaliśmy prawidłową kolejność predykcji (o czym informują krzywe ROC), ale również bardzo dobrą (w większości przypadków) separację wyestymowanych prawdopodobieństw klasy `SC` od `nonSC`.

Regresja logistyczna z regularyzacjami

W zależności od ustalonego parametru regularyzacji λ możemy uzyskać różne sygnatury. Niniejsza praca porównuje wyniki uzyskane dla 3 różnych parametrów λ uzyskanych poprzez optymalizację (a) AUC (ozn. λ_{AUC}), (b) dewiancję (ozn. $\lambda_{Deviance}$) oraz (c) poprawność klasyfikacji (ozn. λ_{Class}). Rysunki 3.4 oraz 3.5 pokazują, że ekstremum funkcji celu (a)–(c) osiągamy przy sygnaturze wielkości zaledwie kilku-kilkunastu z ponad 10 tysięcy (dla *mRNA*) i 20 tysięcy (*methylation*) zmiennych. Najobszerniejszą sygnaturę uzyskujemy przy parametrze λ_{dev}, jest to nawet 40 zmiennych zaś najwęższą: przy maksymalizowaniu AUC. Dokładne informacje o liczbie wybranych zmiennych zawiera poniższa tabela.

λ	methylation.endo (21031)	mRNA.endo (10572)	methylation.meso (17108)	mRNA.meso (10515)
Class	26	12	18	24
AUC	8	3	8	2
Deviance	27	37	29	46

Tabela 4.1: *Wielkość wybranej sygnatury w modelu regresji logistycznej. W nawiasie podano liczbę wszystkich zmiennych w danym zbiorze*

Niezależnie od sporych różnic w liczbie wybieranych zmiennych, możemy zauważyć (wykresy słupkowe na rysunkach 3.8 i 3.9), że wybrane sygnatury są w pewnym sensie hierarchiczne, tzn. cechy ustalone przy parametrze λ_{AUC} były również wybierane przy λ_{Class} oraz $\lambda_{Deviance}$. Wykresy te informują dodatkowo, że współczynniki przy tych samych zmiennych ustalonych przez λ_{AUC}, λ_{Class} oraz $\lambda_{Deviance}$ mają jednakowy co do znaku wpływ na decyzję modelu, co jest potwierdzeniem oczekiwań.

Naturalnym jest, że mogąc zbudować model równie skuteczny na mniejszej liczbie zmiennych, co na większej liczbie, lepszym wyborem jest model zależny od mniejszej liczby zmiennych. Wydawałoby się zatem, że sygnatury wyznaczone przez λ_{AUC} są wystarczające i nie ma potrzeby brać szerszej sygnatury. Jednakże rysunek 3.10 ilustruje, że predykcje klas `SC` i `nonSC` dla sygnatury ustalonej przez λ_{AUC} nie zawsze są dość dobrze rozdzielone. Ustalenie sygnatury poprzez dewiancję zapewniało najdokładniejszą estymację prawdopodobieństw.

Drzewa losowe

Zaprezentowane wyniki pokazują, że *oversampling* nie wprowadza tak znaczącej różnicy w porównaniu ze zwykłym lasem losowym jak próbkowanie *downsampling* zastosowane w procedurze BRF. Drzewa budowane na modelach BRF były wyraźnie mniejsze na wszystkich zbiorach. Ponadto *downsampling* przyczynił się do wyzerowania (w sensie wartości MDG) większej liczby cech na zbiorach bardziej niezrównoważonych.

Głębokość budowanych drzew była różna na różnych podzbiorach: platforma *methylation* wymagała drzew bardziej rozbudowanych niż platforma *mRNA* (rysunek 3.6). Tym niemniej nie były to drzewa bardzo głębokie. Na platformie *mRNA* zdarzały się modele będące samym korzeniem (tzn. o rozmiarze drzewa = 2, odpowiadającym dwóm węzłom decyzyjnym) przeważnie jednak decyzja padała przy 3 lub 4 liściach. Na platformie *methylation* najmniejsze drzewa zawierały 3 liście, najczęściej jednak budowano modele o 5 i 6 liściach.

Sygnatura

Zarówno RF jak i LR doprowadziły do zupełnej eliminacji (wyzerowania) przeważającej liczby zmiennych. Jak wynika z tabeli 4.1, możemy mówić o wyzerowaniu (w sensie wielkości współczynnika przy zmiennej) ponad 99% zmiennych w modelu LR zaś drzewa losowe eliminują - w znaczeniu kryterium Giniego - ponad połowę wszystkich cech (zgodnie z rysunkiem 3.7b).

Można zatem zauważyć, iż pod względem wyboru sygnatury metoda LR wypada lepiej. Użycie regularyzacji LASSO pozwala na wyraźne oddzielenie *ziaren od plew*, podczas gdy RF pozostawia wiele zmiennych z nikłą wartością MDG (na przykładzie 3.7a). Tym niemniej użycie kryterium Giniego do policzenia spadku różnorodności klas w poszczególnych węzłach pozwala wyznaczyć ranking najbardziej wpływowych zmiennych. Kierując się takim rankingiem możemy wybrać n najlepszych cech i powtórnie zbudować model lasów losowych z uwzględnieniem tylko tych cech. Należy przy tym zwrócić uwagę na to, by próbka walidacyjna nie była widziana przy selekcji zmiennych oraz w budowanym modelu.

Dodatkowo możemy zauważyć, iż n najlepszych zmiennych (przy $n = 50$ na rysunkach 3.8 oraz 3.8) wskazanych przez RF zawiera zmienne wybrane przez LR. Najlepsze pokrycie wspólnych cech znajdujemy na zbiorze *methylation* – MESO. Nie zawsze jednak zmienna uznana za najlepsza przez drzewo znajduje swoje uznanie w sygnaturze wybranej przez LR.

Przeuczenie

Przeprowadzając badanie klasyfikatorów zwrócono szczególną uwagę, aby próba walidacyjna nie brała udziału podczas selekcji zmiennych i budowania modelu. Jednakże wyniki krzywych ROC oraz rozkładów predykcji (rysunki 3.12 oraz 3.10) nasuwają wątpliwość, czy udało się modele uchronić przed przeuczeniem. Eksperyment z permutacją wektora odpowiedzi (rysunek 3.13) upewnił nas, że modele LR zachowują się w sposób oczekiwany: AUC na zbiorze walidacyjnym uzyskane z modelu o spermutowanych etykietach oscylują wokół wartości 0.5.

Zaskakujące rezultaty uzyskujemy dla modeli RF: w przeciwieństwie do modeli liniowych, AUC dla lasów potrafi średnio co drugi raz wynosić więcej niż 0.75. Dla modelu zbudowanego na zbiorze `methylation.meso` mediana AUC wynosi 0.8 i jednocześnie

wartości w rozkładzie dewiancji są dużo mniejsze, niż wartości dewiancji w innych modelach. Wskazywałoby to na bardzo dobre nauczenie się modelu na losowych danych. Możemy dodatkowo zauważyć, że im wyższy wskaźnik niedopasowania obiektów, tym większą mamy dewiancję. Może stąd wynikać, iż w skutek permutacji na niezbilansowanych danych, model nauczył się pewnej informacji z klasy nadreprezentowanej. *Oversampling* oraz *downsampling* stabilizują to zachowanie, przy czym to *downsampling* - zastosowany w metodzie BRF - zwraca większą wartość dewiancji na spermutowanym wektorze odpowiedzi.

Klasyfikacja danych nowotworowych

Klasyfikacja na danych nowotworowych, pochodzących w całości ze zbiorów TCGA pokazała, że problem wrażliwości klasyfikatorów na eksperyment, z którego pozyskano dane, udało się wyeliminować: w przeciwnym wypadku uzyskalibyśmy jednolite predykcje bliskie 0, jako że w klasie *porażka* uwzględniono wiele obserwacji kontrolnych pochodzących z TCGA.

Wyniki uzyskane z klasyfikatorów RF i LR (dla trzech różnych sygnatur) są w większości zbieżne. Zgodnie z uzyskanymi predykcjami możemy wnioskować, że najbardziej macierzyste nowotwory to COAD i READ (mediana uzyskanych predykcji powyżej 0.5) oraz OV (opis skrótów w dodatku A). Ponadto u części pacjentów (ok. 20%) z nowotworem BRCA występuje bardzo wysokie podobieństwo do komórek macierzystych, wobec stosunkowo niskiej mediany predykcji w tej grupie.

Dodatki

Dodatek A

Dane TCGA

	Nazwa	Nowotwór	Polska nazwa	Liczba obserwacji	Typ tkanki
1	BRCA	Breast invasive carcinoma	Nowotwór piersi	1098	mezoderma (MESO)
2	COAD	Colon adenocarcinoma	Rak jelita grubego	460	endoderma (ENDO)
3	GBM	Glioblastoma multiforme	Glejak wielopostaciowy	613	ektoderma (ECTO)
4	KIRC	Kidney renal clear cell carcinoma	Rak jasnokomórkowy nerki	537	mezoderma (MESO)
5	KIRP	Kidney renal papillary cell carcinoma	Rak brodawkowaty nerki	323	mezoderma (MESO)
6	LGG	Brain Lower Grade Glioma	Glejak niższych rzędów	516	ektoderma (ECTO)
7	LUAD	Lung adenocarcinoma	Gruczolakorak płuca	585	endoderma (ENDO)
8	LUSC	Lung squamous cell carcinoma	Rak kolczystokomórkowy płuca	504	endoderma (ENDO)
9	OV	Ovarian serous cystadenocarcinoma	Rak jajnika	602	mezoderma (MESO)
10	READ	Rectum adenocarcinoma	Rak odbytnicy	171	endoderma (ENDO)
11	UCEC	Uterine Corpus Endometrial Carcinoma	Rak dróg moczowych	560	mezoderma (MESO)

Tabela A.1: *Dane TCGA w podziale na grupy ENDO, ECTO i MESO. Tabelka sporządzona została w oparciu o informacje zawarte na stronie http://gdac.broadinstitute.org/. Liczba obserwacji dotyczy danych oryginalnych i nie musi pokrywać się z danymi w pakiecie* `RTCGA`

Dodatek B

Kod w R

Generowanie wyników

Ustalenie zmiennych środowiskowych

```
library(RStemnessScorer)
library(gridExtra)
## sciezka do danych i metadanych PCBC.
PCBC_DIR <- "~/RProjects/PCBC_data/"
RESULT_PATH <- "~/RSS_results"
### funkcje normalizacyjne
id <- function(x) x
norm_rank <- function(x) rank(x, na.last = "keep")/length(x)
rank2 <- function(x) rank(x, na.last = "keep")
# FUN_NORM <- c('rank2', 'id', 'norm_rank')
FUN_NORM <- c('norm_rank')
PLATFORMS <- c('methylation', 'mRNA')
TCGA_GROUPS <- c('endo', 'meso')
COMBINATIONS <- c(outer(c(outer(PLATFORMS, TCGA_GROUPS, paste)),
                        FUN_NORM, paste))
COMBINATIONS <- gsub('␣', '.', COMBINATIONS)
```

Przygotowanie danych

```
concat_data <- function(pcbc.dir, platform, group){
  TCGA <<- RTCGATumorNormal(platform, group)
  PCBC.data <<- loadPCBC(pcbc.dir, data=platform,
                         meta_class="Diffname_short")
  PCBC.data <<- RStemnessScorer:::nonNAdata(PCBC.data)
  TCGA$data <<- RStemnessScorer:::nonNAdata(TCGA$data)
  features <<- intersect(colnames(TCGA$data), colnames(PCBC.data))
  PCBC.data <<- PCBC.data[, `:=`((features), lapply(.SD, FUN)),
                          .SDcols = features]
  TCGA$data <<- TCGA$data[, `:=`((features), lapply(.SD, FUN)),
                          .SDcols = features]
  pcbc <<- PCBC.data[, c("Diffname_short", features),
                     with = FALSE]
  TCGA$data <<- TCGA$data[,c(key(TCGA$data),"cancer",features),
                          with=FALSE]
}
```

Generowanie wszystkich wyników

```
for(c in COMBINATIONS){
  platform <- strsplit(c, '\\.')[[1]][1]
  group <- strsplit(c, '\\.')[[1]][2]
  fun <- strsplit(c, '\\.')[[1]][3]
  PATH <- file.path(RESULT_PATH,
                    paste0(platform,'.', group, '.', fun))
  if(!dir.exists(PATH)) dir.create(PATH)
  FUN <- match.fun(fun)
  ## przygotuj dane
  concat_data(PCBC_DIR, platform, group)

  ## Podzial na zbi r treningowy i walidacyjny
  signature <- features
  learn <- setupLearningSets(pcbc, TCGA$data[TCGA$normal, ],G='SC',
                             signature = features, cutoff = 0.7)
  save(learn, file = file.path(PATH, 'learn.rda'))

  ## Modele: regresja logistyczna i lasy losowe
  objectiveFun <- c("Class", "AUC", "Deviance")
  models <- lapply(objectiveFun, function(f)
    buildScorer(learn$train$X,
                learn$train$Y, model = "LR", cv.measure=tolower(f),
                intercept = TRUE, standardize = FALSE, njob = 3))
  names(models) <- objectiveFun
  models[['RandomForest']] <- buildScorer(
    learn$train$X, y, model = "RF", ntree = 5000, njob = 3, proximity=TRUE)
  models[['RandomForest_brf']] <- buildScorer(
    learn$train$X, y, model = "RF", ntree = 5000, njob = 3,
    sampsize=rep(min(table(y)), nlevels(y)), proximity=TRUE)
  oversampling <- function(x,y){
    tab <- table(y)
    ratio <- floor(max(tab)/min(tab) - 1)
    min_class <- names(which.min(tab))
    ind = y == min_class
    i = 0
    x2 = x
    y2 = y
    while(i<(ratio-1)){
      x2 = rbind(x2, x[ind,])
      y2 = factor(c(as.character(y2), as.character(y[ind])))
      i = i+1
    }
    return(list(x=x2, y=y2))
  }
  ovsmpl <- oversampling(learn$train$X, learn$train$Y)
  models[['RandomForest_ovrsmpl']] <- buildScorer(
    ovsmpl$x,ovsmpl$y, model="RF", ntree=5000, proximity=TRUE, njob = 3)
  save(models, file = file.path(PATH, 'models.rda'))

  ## predykcja na zbiorze testowym
  df <- data.frame()
  for (of in names(models)) {
    scores <- scorer(learn$test$X, models[[of]])
    s <- split(scores, learn$test$Y)
    df <- rbind(df, data.frame(melt(s), f = of))
  }
  colnames(df)[2] <- "class"
  tumor <- testTumor(TCGA$data[TCGA$tumor, ], models)
  save(df, file = file.path(PATH, 'valid_scores.rda'))
  save(tumor, file = file.path(PATH, 'tumor.rda'))
}
```

Y-permutacja

```
permutationTest <- function( learn ){
  y = sample(learn$train$Y)
  objectiveFun <- c("Class","AUC","Deviance")
  models <- lapply(objectiveFun, function(f)
    buildScorer(learn$train$X, y, model='LR',
                cv.measure=tolower(f),
```

```
                intercept=TRUE, standardize=FALSE,
                nfolds=floor(dim(learn$train$X)[1]/3),
                njob = 4))
  names(models)[1:3] <- objectiveFun

  rf <- buildScorer(learn$train$X,y, model="RF",ntree=1000,
                    njob = 3, keep.forest=TRUE)
  ovsmpl <- oversampling(learn$train$X, y)
  rf_o <- buildScorer(ovsmpl$x,ovsmpl$y, model="RF",ntree=1000,
                      njob = 3, keep.forest=TRUE)
  brf <- buildScorer(learn$train$X,y, model="RF",ntree=1000,
                     njob = 3, keep.forest=TRUE,
                     sampsize=rep(min(table(y)), nlevels(y)))
  models[['RandomForest_unbalanced']] <- rf
  models[['RandomForest_oversampling']] <- rf_o
  models[['RandomForest_balanced']] <- brf
  df <- data.frame()
  for( of in names(models) ){
    scores <- scorer(learn$test$X, models[[of]])
    s <- split(scores, learn$test$Y)
    df <- rbind(df, data.frame(melt(s), f=of))
  }
  colnames(df)[2] <- 'class'
  return(list(models = models, pred=df, y = y))
}

n <- 100

for( c in COMBINATIONS ){
  print(c)
  nm <- gsub('␣', '.', c)
  load(file.path(RESULT_PATH, nm, 'learn.rda'))
  test100 <- as.list(numeric(n))
  pb <- txtProgressBar(min = 0, max = n, style = 3)
  for( i in 1:n ){
    setTxtProgressBar(pb, i)
    out <- permutationTest(learn)
    test100[[i]] <- out
  }
  save(test100, file=file.path(RESULT_PATH, 'YRANDOM',
                               paste0(nm,'.rda')))
}
```

Generowanie rysunków

```
read_models <- function(which = '.' ){
  lapply(COMBINATIONS, function(c){
    load(file.path(RESULT_PATH, c, paste0('models.rda')))
    return(models[grepl(which, names(models))])
  }) -> modele
  names(modele) <- COMBINATIONS
  return(modele)
}
```

Ścieżki współczynników oraz wartości poszczególnych funkcji celu dla różnych parametrów regularyzacji LASSO.

```
M <- read_models('AUC|Dev|Class')

### sciezki
par(mfrow=c(4,2),  mai = c(0.5, .4, .6, 0.1))
c2 <- c(1,3)
for(m in names(M)[c2]){
  plot(M[[m]][['Class']]$glmnet.fit, 'lambda', mgp=c(1.4,.3,.0))
  title(line = 3, main=m)
}
par(mai = c(0.5, .4,.6, 0.1))
```

```
for(m in names(M)[c2]) plot(M[[m]][['Class']], mgp=c(1.4,.3,.0))
mtext(text='Class', line=-25, outer=TRUE)
for(m in names(M)[c2]) plot(M[[m]][['Deviance']], mgp=c(1.4,.3,.0))
mtext(text='Deviance', line=-48, outer=TRUE)
for(m in names(M)[c2]) plot(M[[m]][['AUC']], mgp=c(1.4,.3,.0))
mtext(text='AUC', line=-70, outer=TRUE)
dev.off()

### sciezki
par(mfrow=c(4,2),  mai = c(0.5, .4, .6, 0.1))
c2 <- c(2,4)
for(m in names(M)[c2]){
  plot(M[[m]][['Class']]$glmnet.fit, 'lambda', mgp=c(1.4,.3,.0))
  title(line = 3, main=m)
}
par(mai = c(0.5, .4,.6, 0.1))
for(m in names(M)[c2]) plot(M[[m]][['Class']], mgp=c(1.4,.3,.0))
mtext(text='Class', line=-25, outer=TRUE)
for(m in names(M)[c2]) plot(M[[m]][['Deviance']], mgp=c(1.4,.3,.0))
mtext(text='Deviance', line=-48, outer=TRUE)
for(m in names(M)[c2]) plot(M[[m]][['AUC']], mgp=c(1.4,.3,.0))
mtext(text='AUC', line=-70, outer=TRUE)
dev.off()
```

Udział liczby drzew o danym rozmiarze w poszczególnych klasyfikatorach

```
RF <- read_models('Forest')

lapply(RF, function(rfs) {
  lapply(names(rfs), function(rf.type){
    tree.size <- table(treesize(rfs[[rf.type]]))
    liczba.drzew <- reshape2::melt(tree.size, value.name = 'Liczba.drzew')
    return(data.frame(liczba.drzew, model=rf.type))
  }) -> dfs
  return(do.call(rbind, dfs))
}) -> DF
DF <- reshape2::melt(DF, id = c('Var1', 'Liczba.drzew'))
colnames(DF)[c(2,4,5)] <- c('Liczba␣drzew', 'Probkowanie', 'Zbiory␣danych')
DF$Probkowanie <- ifelse(DF$Probkowanie == 'RandomForest', 'unbalanced␣RF',
                         ifelse(grepl('ovrsmpl', DF$Probkowanie), 'oversampling',
                                'balanced␣RF'))
DF$Probkowanie <- factor(DF$Probkowanie,
                         levels = c('unbalanced␣RF', 'oversampling', 'balanced␣RF'))
ggplot(data = DF) +
  facet_grid(.~Probkowanie) +
  geom_bar(aes(x = Var1, y = `Liczba drzew`, fill = `Zbiory danych`), stat = 'identity') +
  scale_fill_manual(values=c('lightsteelblue','lightslategrey','mistyrose3','lightpink4')) +
  scale_x_discrete(name ="Rozmiar␣drzewa", limits=unique(DF$Var1))
```

Miara różnorodności Giniego w drzewach losowych

```
RF <- read_models('Forest')
lapply(RF, function(rf){
  reshape2::melt(sapply(rf, function(x) table(importance(x)==0)))
  }) -> GG
GG <- reshape2::melt(GG)
levels(GG$Var2) <- c('RF', 'BRF', 'RF_O')
GG$Var1 <- ifelse(GG3$Var1, 'tak', 'nie')   # Var1 MDG==0
GG$L1 <- gsub('.norm_rank', '', GG$L1)
GG$L1 <- gsub('\\.', '\n', GG$L1)
library(dplyr)
GGb <- GG %>% group_by(L1,Var2) %>% mutate( mn = value/sum(value)*100)
ggplot(GGb) + facet_grid(. ~ L1 ) +
  geom_bar(aes(x=Var2, y=mn, fill=Var1), stat='identity') +
  labs(x='typ␣modelu', y='%␣liczby␣zmiennych', fill='MDG␣==␣0?') +
  scale_fill_manual(values = c( "#E6AF00", "#65B4AE")) +
  theme(legend.position='bottom') -> gg1

ggplot() +
  geom_boxplot(aes(x=as.factor(1),y=importance(RF[[3]][[1]]))) +
```

```
  labs(y='Wartosci MDG', x='', title='') +
  theme(plot.title = element_text(hjust=.5),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  coord_flip() -> gg2
```

Sygnatura genetyczna

```
signature_analysis <- function(models, glmnet_coefs, nm){
  rf <- models$RandomForest_undrsmpl
  rf_plot <- plotVarImp(rf, n.var = 50, size=2) +
    theme(axis.text.x = element_text(angle = 45, size=7, hjust = 1),
          axis.title.x = element_text(size=10))
  glm_plot <- glmnet_coefs +
    theme(legend.key.size = unit(.8, 'lines'), legend.title = element_blank()) +
    theme(legend.position = 'bottom')
  common <- intersect(levels(glm_plot$data$signature), levels(rf_plot$data$Features))
  ind <- levels(rf_plot$data$Features) %in% common
  rf_plot_settings <- theme(
    axis.text.x = element_text(color=ifelse(ind, 'brown3', 'gray40'),
                               face=ifelse(ind, 'bold', 'plain'),
                               size=10),
    axis.title.x = element_blank())
  ind <- levels(glm_plot$data$signature) %in% common
  glm_plot_settings <- theme(
    axis.text.x = element_text(angle=45, hjust=1,
                               color=ifelse(ind, 'brown3', 'gray40'),
                               face=ifelse(ind, 'bold', 'plain'),
                               size=10),
    axis.title.x = element_blank())
  out <- arrangeGrob(rf_plot+rf_plot_settings+coord_flip(),
                     glm_plot+glm_plot_settings, nrow=2,
                     top = nm, heights = c(1,2))
}
lapply(names(M), function(i) signature_analysis(M[[i]], GG2[[i]], i)) -> W
grid.arrange(W[[1]])
```

Krzywe ROC

```
lapply(COMBINATIONS, function(c){
  load(file.path(RESULT_PATH, c, 'valid_scores.rda'))
  df$class <- ifelse(df$class == 'SC', 'SC', 'nonSC')
  return(df)
}) -> VS
names(VS) <- COMBINATIONS
D <- reshape2::melt(VS)
D[['Typ modelu']] <- ifelse(grepl('Forest', D$f), 'nieliniowy', 'liniowy')
levels(D$f) <- c("Class", "AUC", "Deviance", "RF", "BRF", "RF+oversampling")
colnames(D)[4] <- 'predykcje'
by(D, INDICES = list(D$f, D$L1), FUN = function(d){
  ROC <- pROC::roc(d$class, d$predykcje)
  FPR <- 1 - ROC$specificities
  TPR <- ROC$sensitivities
  n <- length(FPR)
  return(data.frame(fpr = FPR,
                    tpr = TPR,
                    model = rep(d$f[1], n),
                    platforma = rep(strsplit(d$L1[1], '\\.')[[1]][1], n),
                    grupa = rep(strsplit(d$L1[1], '\\.')[[1]][2], n)))
}) -> dane2
dane2 <- do.call(rbind, dane2)
g <- ggplot(dane2, aes(fpr, tpr, color = model)) +
  facet_grid( platforma ~ grupa) +
  theme(plot.title = element_text(hjust=.5)) +
  geom_path(size = .9, alpha = .9) +
  labs(x = "FPR", y = "TPR", title = 'Krzywe ROC')
```

Rozkład predykcji 1 i 2

```
D$L1 <- gsub('.norm_rank','', D$L1)
D$L1 <- gsub('\\.','\n', D$L1)
levels(D$f) <- c('Class', 'AUC', 'Deviance', 'RF', 'BRF', 'RF_0')
ggplot(D) + geom_histogram(aes(predykcje, fill=class), bins=50) +
  scale_fill_manual(values=c("darkolivegreen3", "coral3")) +
  labs(title='Rozklad predykcji', x='') +
  facet_grid(L1 ~ f, scales='free_y') +
  theme(plot.title = element_text(hjust=.5, size=15),
        legend.key.size = unit(.5, 'cm'),
        plot.margin=unit(c(0.05,0,0,0), "cm")) +
  guides(fill = guide_legend(title.position="top", title.hjust = 0.5, size=1)
         )-> g1

lapply(names(VS), function(nm){
  D2 <- VS[[nm]]
  D2 <- data.frame( RF = rep(D2$value[D2$f=='RandomForest'], 3),
                    LR = D2$value[!grepl('RandomForest',D2$f)],
                    LR_type = D2$f[!grepl('RandomForest',D2$f)],
                    set = nm)
}) -> out
D2 <- do.call(rbind, out)
D2$set <- gsub('.norm_rank', '', as.character(D2$set))
D2$set <- gsub('\\.', '\n', as.character(D2$set))
theme_set(theme_gray(base_size = 14))
ggplot(D2, aes(x = RF, y=LR, group=LR_type, color=LR_type)) +
  geom_point(size=1) +
  geom_smooth() +
  facet_grid( ~set) +
  labs(x = 'predykcja las wlosowych', y='predykcja\nregresji logistycznej') +
  scale_color_manual(name='Parametr LASSO',
                     values=c("#65B4AE", "orange2", 'plum4')) +
  scale_x_continuous(breaks=seq(0,1,0.2), limits = c(0,1)) +
  coord_fixed() +
  theme(plot.title = element_text(hjust=.5),
        axis.text.x = element_text(angle=45),
        legend.position='bottom',
        legend.box = 'horizontal',
        legend.title=element_text(size=8),
        legend.key.size = unit(.5, 'cm')) -> g2
```

Y-permutacja

```
auc <- pROC::auc
dev <- function(y, p) return( -2*sum(mapply(function(yi, pi)
  yi * log(pi) + (1-yi) * log(1-pi), y, p)) )
hamming <- function(n) return((n[1,2]+n[2,1] ) / sum(n))

plotTest <- function(file1 = 'mRNA.endo.norm_rank.rda',
                     file2 = 'mRNA.endo.norm_rank'){
  require(dplyr)
  load(sprintf('%sPERMUTACJE/%s', RESULT_PATH, file1))
  load(sprintf('%s%s/learn.rda', RESULT_PATH, file2))
  lapply( test100, function(t){
    scores <- data.frame()
    for( of in names(t$models) ){
      s <- scorer(learn$test$X, t$models[[of]]) %>% split(., learn$test$Y)
      scores <- rbind(scores, data.frame(melt(s), f=of))
    }
    colnames(scores)[2] <- 'class'
    levels(scores$f) <- c("Class", "AUC", "Deviance",
                          "RF", "RF+\noversampling","BRF")
    AUC <- sapply(split(scores, scores$f), function(x) auc(x$class,x$value))
    DEV <- sapply(split(scores, scores$f), function(x) dev(ifelse(x$class=='SC',1,0),x$value) )
    return(list(auc = as.data.frame(AUC), dev = as.data.frame(DEV), pokrycie = table(t$y, learn$train
  }) -> auce1
  auc_phi_100 <- data.frame( t(do.call(cbind, lapply(auce1, function(x) x$auc))),
                             phi = sapply(auce1, function(x) hamming(x$pokrycie)) )
  dev_phi_100 <- data.frame( t(do.call(cbind, lapply(auce1, function(x) x$dev))),
                             phi = sapply(auce1, function(x) hamming(x$pokrycie)) )
  D <- data.frame(
```

```
    rbind(auc_phi_100, dev_phi_100),
    measure=c(rep('Rozklad␣miary␣AUC', nrow(auc_phi_100)),
              rep('Rozklad␣dewiancji', nrow(dev_phi_100))))
  colnames(D) <- gsub('\\.', '\n', colnames(D))
  d <- reshape2::melt(D, id.vars=c('phi', 'measure'), variable.name='Model')
  d1 <- data.frame(d, `typ modelu`=ifelse(grepl('RF', d$Model), 'Random␣Forest','Logistic␣Regression'))
  ggplot(data=d1) +
    geom_boxplot(aes(x=mean(d1$phi), y=value, fill = typ.modelu), alpha=.65) +
    scale_fill_manual(values=c("#E6AF00", "#65B4AE")) +
    geom_point(aes(y=value,x=phi),shape='o',size=3,color='brown', alpha=.7) +
    facet_grid( Model~ measure, scales='free_x' ) +
    coord_flip() +
    labs(y='', x='Wspolczynnik␣niedopasowania␣obiekt w') +
    theme(legend.position = 'bottom') -> g1
  return(g1)
}

for(s in gsub('.norm_rank', '', COMBINATIONS)){
  g <- plotTest(file1 = sprintf('%s.norm_rank.rda', s), file2 = sprintf('%s.norm_rank', s))
  assign(s, g)
}
```

Predykcja dla komórek nowotworowych

```
lapply(COMBINATIONS, function(c){
  load(file.path(RESULT_PATH, c, 'tumor.rda'))
  return(tumor + theme(axis.text.x = element_text(angle=45, vjust=0.5)) + ggtitle(c))
}) -> TU
names(TU) <- COMBINATIONS
D <- lapply(TU, function(x) x$data)
D <- reshape2::melt(D)
temp <- do.call(rbind, strsplit(D$L1, '\\.')); D$L1 <- NULL
D <- data.frame(D, temp)
colnames(D) <- c('Nowotwor', 'model' ,'variable', 'predykcje', 'platforma', 'Grupa␣TCGA')
D$`Grupa TCGA` <- toupper(D$`Grupa TCGA`)
levels(D$model) <- c("Class", "AUC", "Deviance", "RF", "BRF", "RF+oversampling")
theme_set(theme_gray(base_size = 18))
out <- by(data = D, INDICES = list(D$`Grupa TCGA`), FUN = function(m) {
  m <- droplevels(m)
  g <- ggplot(m) + facet_grid(platforma ~ `Grupa TCGA`) +
    geom_boxplot(aes(x=model, y=predykcje, fill=Nowotwor)) +
    labs(y = '\t␣non␣SC␣<----␣\t\t␣prawdopodobienstwo␣\t\t␣---->␣SC␣\t', x = '') +
    theme(axis.text.x = element_text(angle=0, vjust=0.5),
          legend.position = 'right') +
    scale_fill_discrete(name = sprintf("Nowotwory\nz␣grupy␣%s",m[1,6]))
})
out <- do.call(grid.arrange, out)
```

Spis rysunków

Spis tabel

Bibliografia

[1] Nicolas Bacaer. Verhulst and the logistic equation (1838). In Henry Horng-Shing Lu, Bernhard Schölkopf, and Hongyu Zhao, editors, *A Short History of Mathematical Population Dynamics*, chapter 6. 2011.

[2] Leo Breiman. Random forest. *Machine Learning*, 45(1):5–32, 10 2001.

[3] Chao Chen, Andy Liaw, and Leo Breiman. Using random forest to learn imbalanced data. *UC Berkeley Technical Reports*, (666), 2004.

[4] C. Drummond and R. Holte. C4.5, class imbalance, and cost sensitivity: why undersampling beats over-sampling. *Proceedings of the ICML-2003 Workshop: Learning with Imbalanced Data Sets II*, 2003.

[5] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[6] Salomonis et al. Integrated genomic analysis of diverse induced pluripotent stem cells from the progenitor cell biology consortium. *Stem Cell Reports*, 7(1):110–125, 7 2016.

[7] Elana J. Fertig, Ludmila V. Danilova, and Michael F. Ochs. Cancer systems biology. In Henry Horng-Shing Lu, Bernhard Schölkopf, and Hongyu Zhao, editors, *Handbooks of Computational Statistics*, chapter 25. 2011.

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

[9] Leroy Hood and Lee Rowen. The human genome project: big science transforms biology and medicine. *Genome Med.*, 2013.

[10] Bio-Rad Laboratories Inc. Stem cell basics for life science. *Nature Genetics*, (45):1113–1120, 2013.

[11] Friedman Jerome, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

[12] Jacek Koronacki and Jan Ćwik. *Statystyczne systemy uczące się*. Akademicka Oficyna Wydawnicza EXIT, 2 edition, Warszawa 2008.

[13] Marcin Kosinski and Przemyslaw Biecek. *RTCGA: The Cancer Genome Atlas Data Integration*, 2016. R package version 1.2.5.

[14] National Institutes of Health. Fact sheet: Human genome project. 2010.

[15] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[16] J. Weinstein, E. Collisson, G. Mills, K. Shaw, B. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. Stuart. Cancer genome atlas research network. *Nature Genetics*, (45):1113–1120, 2013.

Warszawa, dnia .....

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem: „Analiza metod uczenia maszynowego wykorzystywanych w budowaniu sygnatur genetycznych", której promotorem jest dr hab. inż. Przemysław Biecek prof. nzw. wykonałam samodzielnie, co poświadczam własnoręcznym podpisem.

.....