Drzewa decyzyjne i lasy losowe

Im dalej w las tym więcej drzew!

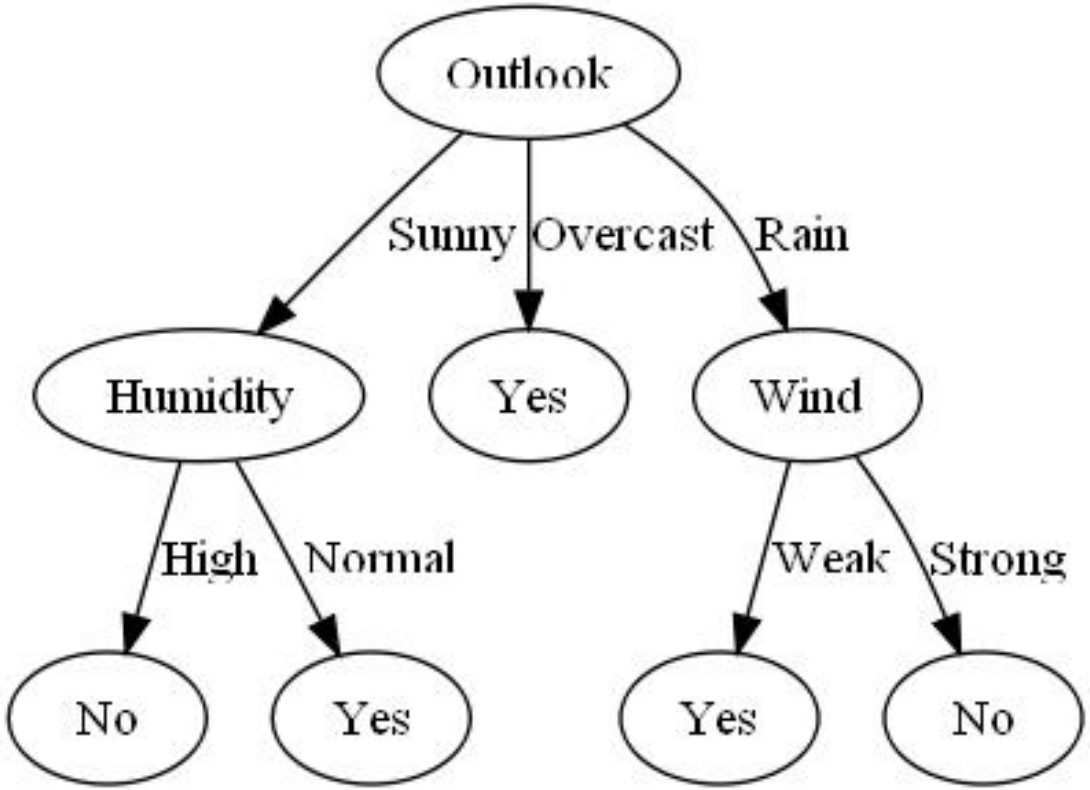
— Marcin Zadroga
https://www.linkedin.com/in/mzadroga/

ML Gdańsk
http://www.mlgdansk.pl/

20 Czerwca 2017
WPROWADZENIE DO MACHINE LEARNING

CZYM JEST DRZEWO?

- Skierowany spójny graf acykliczny
- Pierwszy wierzchołek – korzeń
- Krawędzie – gałęzie
- Węzły końcowe - liście
- Droga – przejście od korzenia wzdłuż kolejnych gałęzi (do liścia)



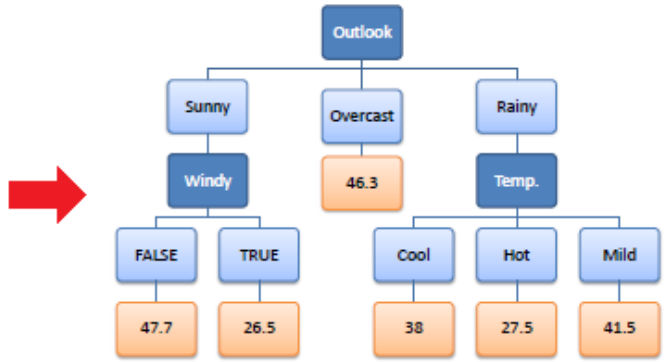
Źródło: https://blogs.msdn.microsoft.com/chrsmith/2009/11/02/awesome-f-decision-trees-part-ii/

ZADANIA DLA DRZEWA

- Zadania klasyfikacji
- Zadania regresji

- Uczenie na podstawie zbioru, czego nie ma w zbiorze tego nie wiemy …

Predictors				Target
Outlook	Temp	Humidity	Windy	Hours Played
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	52
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	35
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Źródło: http://www.saedsayad.com/decision_tree.htm

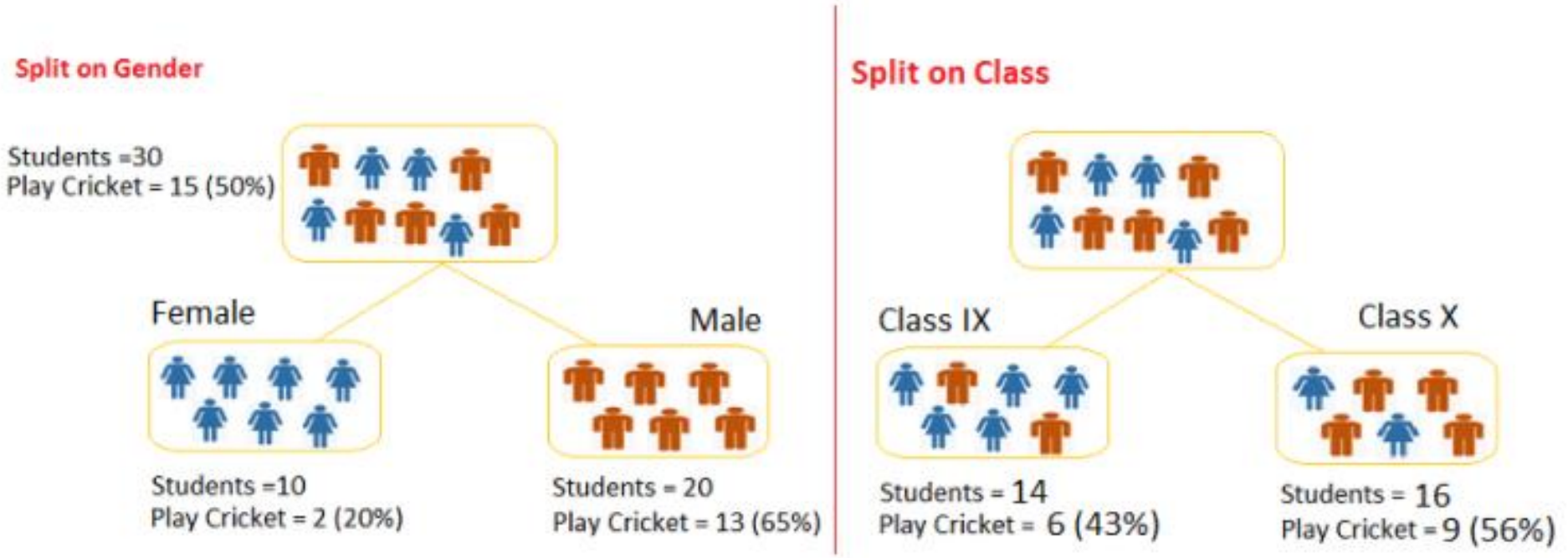
PODZIAŁ DRZEWA

- Podział ma być przeprowadzony tak aby różnorodność próbek docierających do wezłów dzieci była możliwie najmniejsza – podział **lokalnie** optymalny

- Miary różnorodności klas
  - Proporcja błędnych klasyfikacji
  - Indeks Giniego
  - Entropia
- Miarę różnorodności klas w dzieciach węzła obliczamy jako ważoną sumę

- Indeks Giniego i Entropia są bardziej czułe na zmiany rozkładów klas

PRZYKŁAD



Split on Gender:

1. Calculate, Gini for sub-node Female = (0.2)*(0.2)+(0.8)*(0.8)=0.68
2. Gini for sub-node Male = (0.65)*(0.65)+(0.35)*(0.35)=0.55
3. Calculate weighted Gini for Split Gender = (10/30)*0.68+(20/30)*0.55 = **0.59**

Similar for Split on Class:

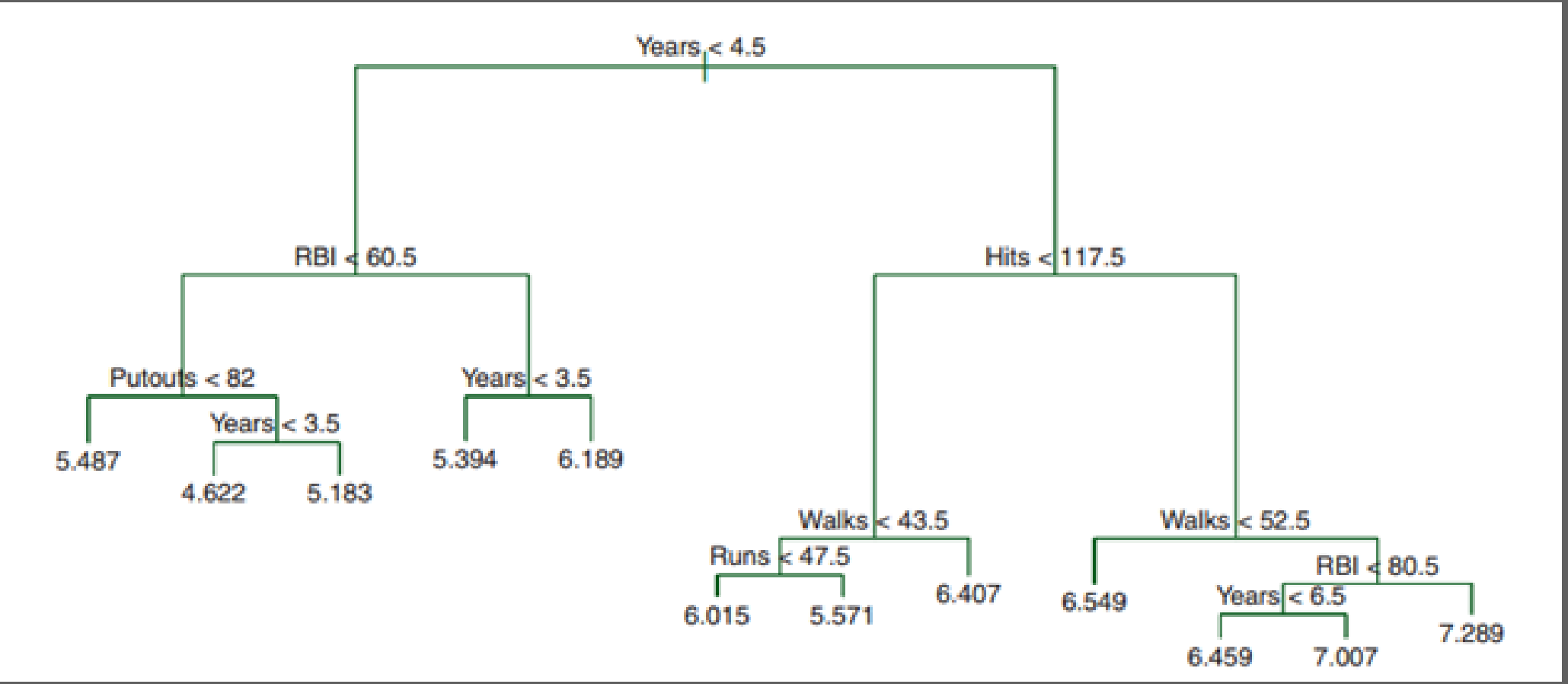
1. Gini for sub-node Class IX = (0.43)*(0.43)+(0.57)*(0.57)=0.51
2. Gini for sub-node Class X = (0.56)*(0.56)+(0.44)*(0.44)=0.51
3. Calculate weighted Gini for Split Class = (14/30)*0.51+(16/30)*0.51 = **0.51**

Źródło: https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

PRZYCINANIE DRZEWA

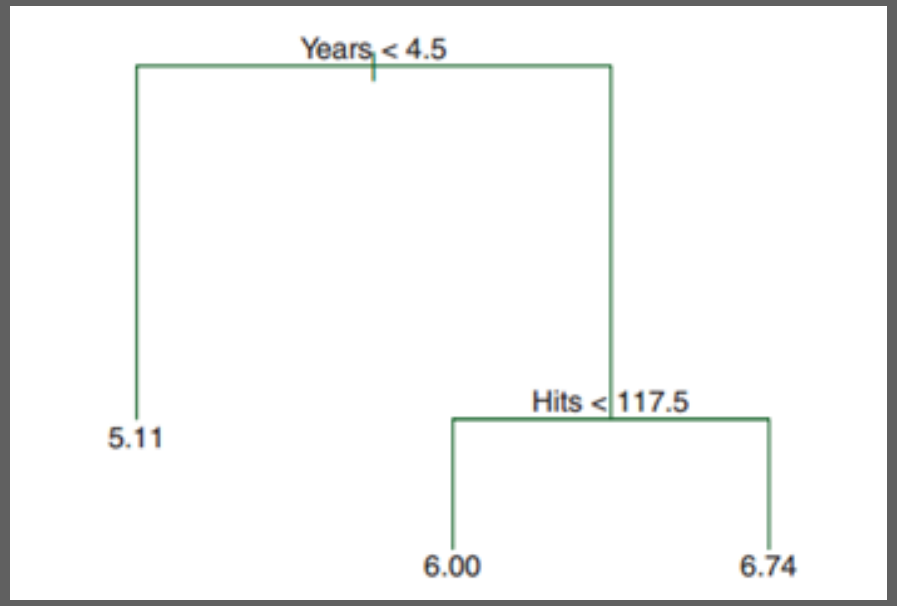
- Drzewo odzwierciedli tylko nasz zbiór – przeuczy się, trzeba je przyciąć!
- Budujemy „pełne” drzewo (w węzłach obserwacje z tej samej klasy)
- Sposoby:
  1. Heurystyka – 5 elementów w liściu i nie idziemy dalej
  2. Heurystyka – Ograniczenie wysokości drzewa
  3. Ucinamy po liściu, tak aby zmniejszyć błąd na zbiorze walidacyjnym
  4. Kryterium kosztu – złożoności $R_\alpha(T) = R(T) + \alpha|T|$, zadanie znalezienia minimum
  5. Algorytm Breimana – wybór optymalnego Alfa (współczynnika złożoności), 1SE

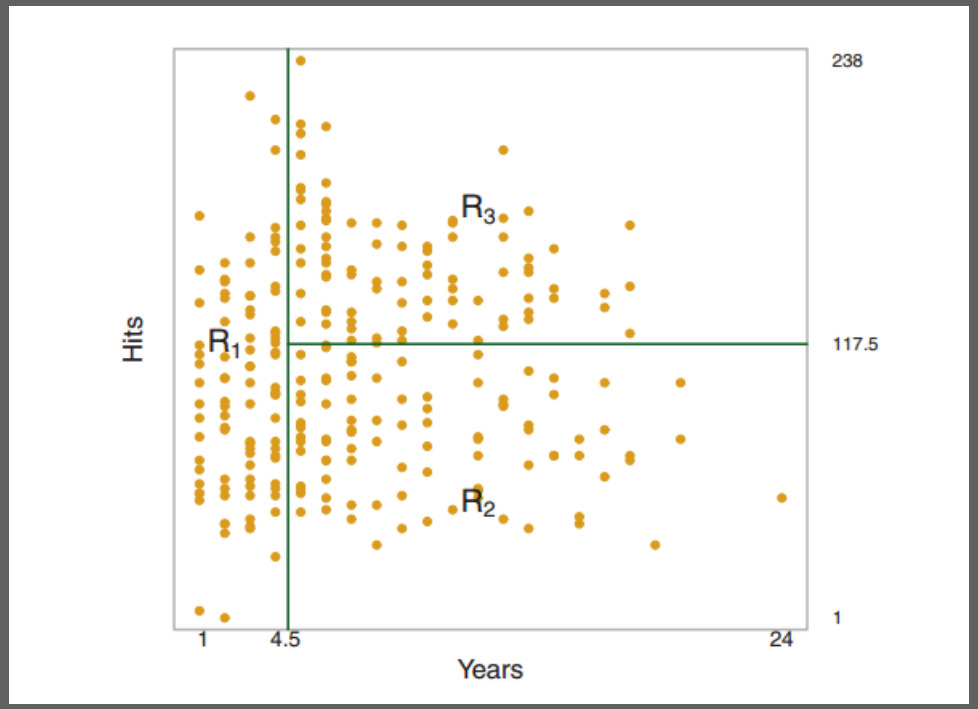
ZADANIE REGRESJI DLA DRZEWA



Źródło: „An introduction to statistical learning" – G. James, D. Witten, T. Hastie, R. Tibshirani

ZADANIE REGRESJI DLA DRZEWA





Źródło: „An introduction to statistical learning" – G. James, D. Witten, T. Hastie, R. Tibshirani

DRZEWA – PLUSY I MINUSY

- Łatwe do wytłumaczenia i zrozumienia – prezentacja dla managerów ;)
- Odwzorowują ludzkie podejście
- Zadanie klasyfikacji możliwe do realizacji bez tworzenia „dummy variables” – łatwo je zastosować bez inżynierii danych

- Duża niestabilność – mała zmiana danych, duża zmiana drzewa
- Podatne na przeuczenie

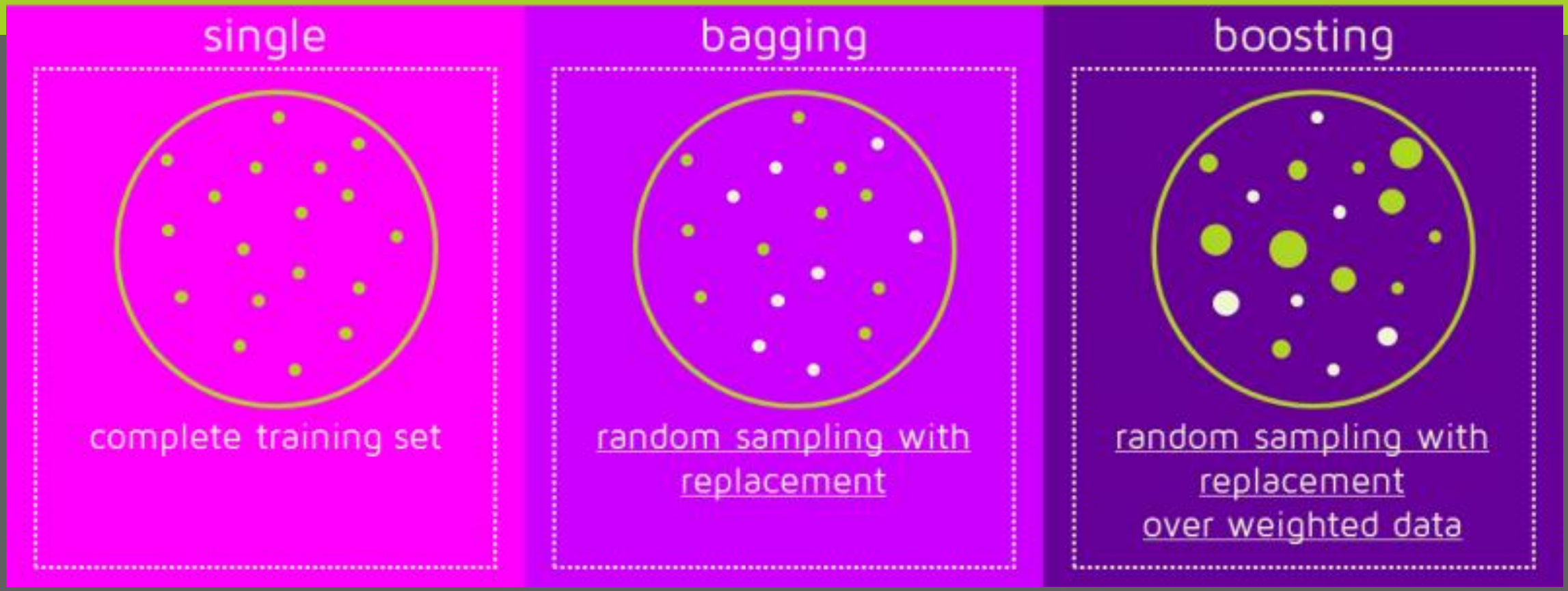
BAGGING

- Mamy dane *C* „lichych” klasyfikatorów (problem z dwiema klasami) – słabi uczniowie, niewiele lepsi niż rzut monetą (0,55)
- Jeżeli *C* jest duże, to większość klasyfikatorów dokona poprawnej klasyfikacji

- Wytrenujmy „pełne” drzewa na **pseudopróbach**, o liczności n < N (Breinman)
  - Losowanie z rokładu 1/n z zwracaniem
- Drzewa „głosują” zwykłą większością

BOOSTING

- Zaproponowana w 1990 przez Schapire'a
- Losowanie pseudoprób (ze zwracaniem), rozkład jednostajny do pierwszej pseudopróby
- Kolejne, rozkład zmienia się, **zwiększamy prawdopodobieństwa wylosowania próby**, która została niepoprawnie zaklasyfikowana przez poprzedni klasyfikator
- (META)Algorytm AdaBoost – dyskretny adaptacyjny boosting, bez losowania



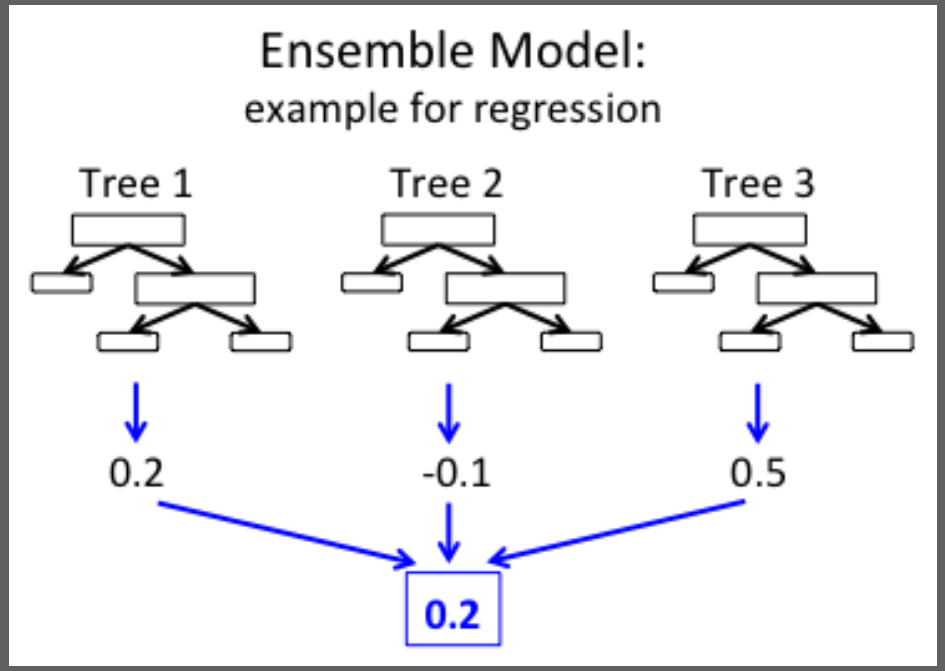
Źródło: https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

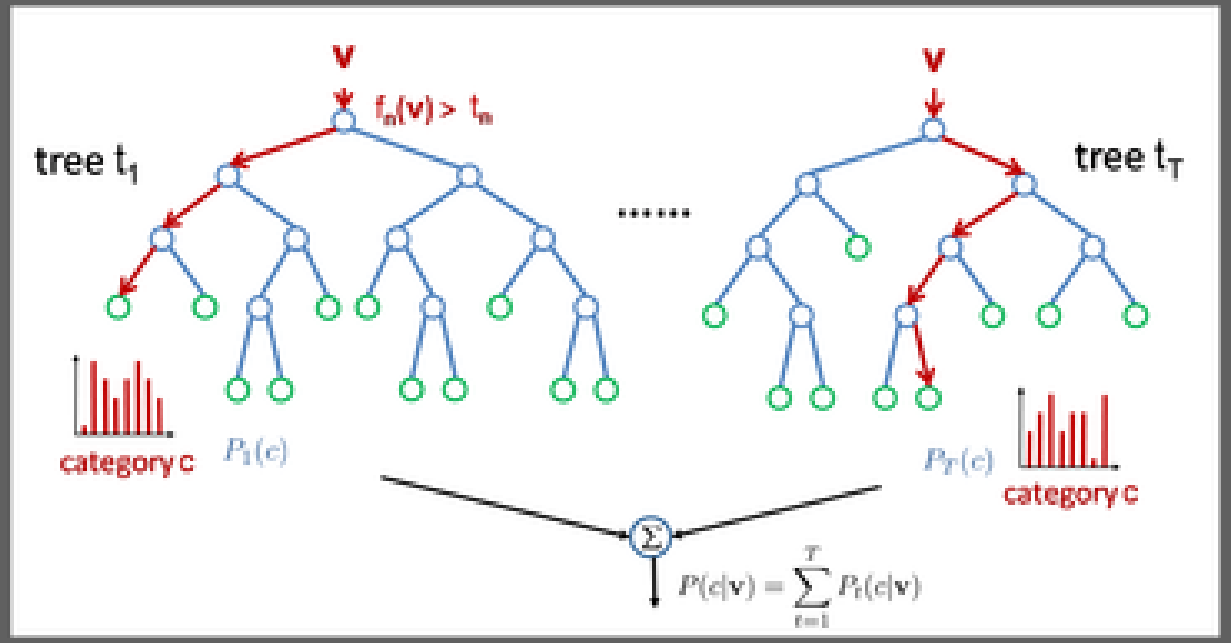
LASY LOSOWE

1. Wylosuj ze zwracaniem *n*-elementową pseudopróbę
2. W każdym węźle (!) wylosuj *m* spośród *p* atrybutów (bez zwracania) i zastosuj wybraną metodę podziału, b. często *m<<p* np. *m=sqrt(p)*
3. Zbuduj drzewo bez przycinania, aż do otrzymania liści z elementami z jednej klasy
4. Zbuduj *C* takich drzew
5. Drzewa „głosują” nad rozwiązaniem, wybór zwykłą większością głosów

Lasy losowe to uogólnienie bagging, gdzie m<>p

LASY LOSOWE - UŻYCIE





Źródło (1): https://databricks.com/blog/2015/01/21/random-forests-and-boosting-in-mllib.html

Źródło (2): http://www.iis.ee.ic.ac.uk/icvl/iccv09_tutorial.html

LASY LOSOWE – PLUSY DODATNIE I PLUSY UJEMNE

+ Dobre wyniki! ☺

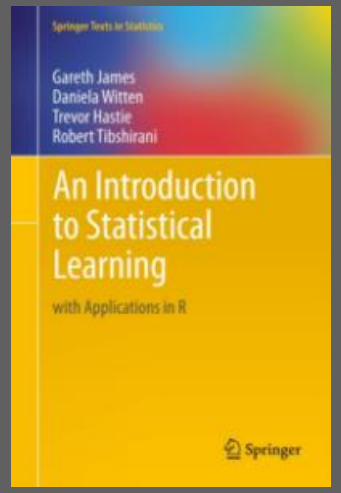
+ Odpowiednie dla dużych zbiorów $m<<p$

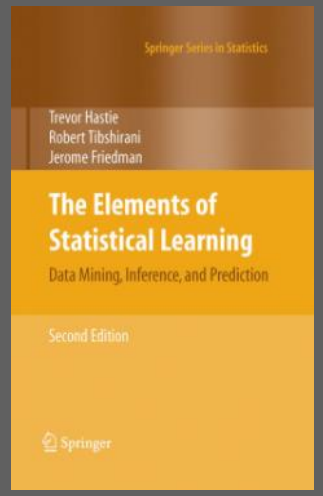
+ „Dają” oszacowanie, które zmienne są ważne (zwiększone prawdopodobieństwo)

- Czasem niestety się przeuczają (dane zaszumione)

LITERATURA

- „Statystyczne systemy uczące się” – J. Koronacki, J. Ćwik
- „An introduction to statistical learning” – G. James, D. Witten, T. Hastie, R. Tibshirani
- „The elements of statistical learning” – T. Hastie, R. Tibshirani, J. Friedman





DZIĘKUJĘ ZA UWAGĘ!