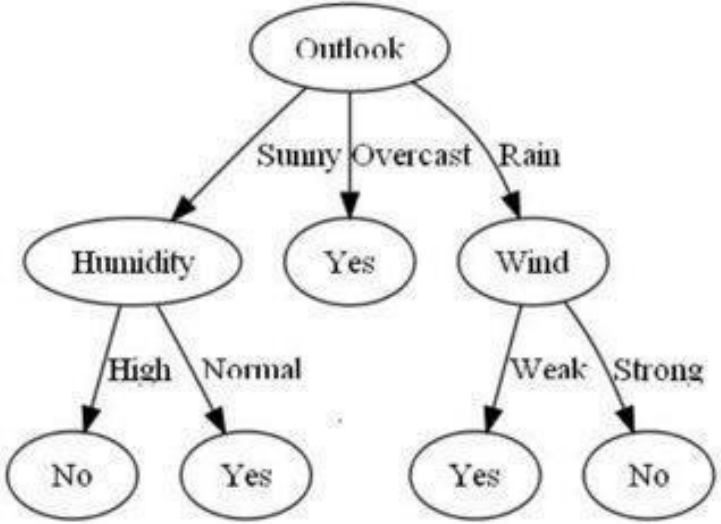
Random Forest

Ale to już było... drzewa decyzyjne

- Proste modele opierające się na podziale danych treningowych przez serię porównań, w taki sposób, aby różnorodność próbek docierających do węzłów dzieci była możliwie najmniejsza.
- Intuicyjne, łatwe do wytłumaczenia. Szybkie i skuteczne.
- Odpowiednie dla problemów regresji i klasyfikacji.
- Wyniki są wyjaśnialne i łatwe do wizualizacji.
- Niestabilne i podatne na przeuczenie.



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

infoShare ACADEMY

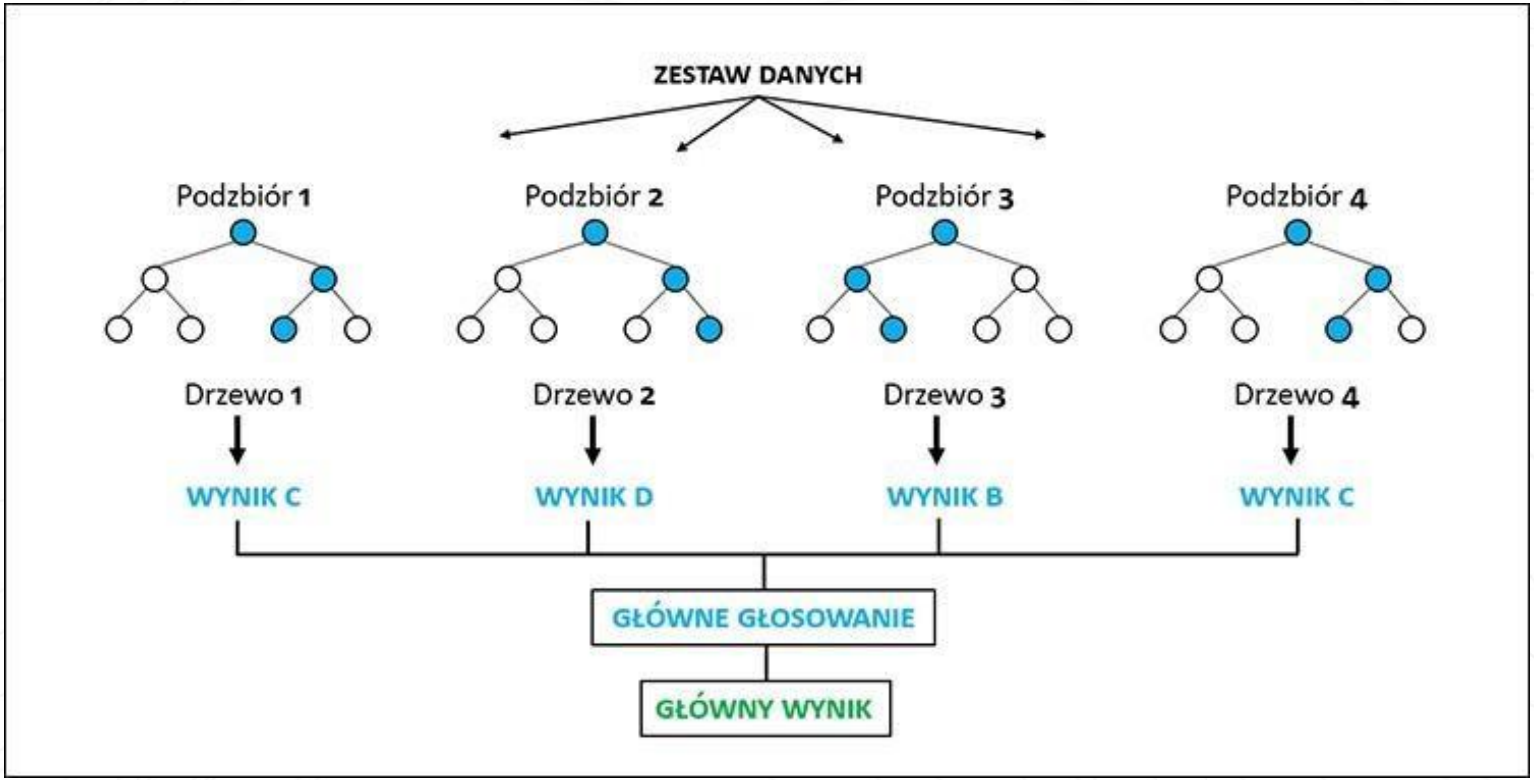
Random Forest

Im dalej w las tym więcej drzew ...

Random Forest

- Random Forest to model typu ensemble (komitet klasyfikatorów) czyli model składający się ze zbioru słabszych modeli, których wyniki są następnie przetwarzane (uśredniane lub przeliczane) w celu stworzenia modelu silnego.

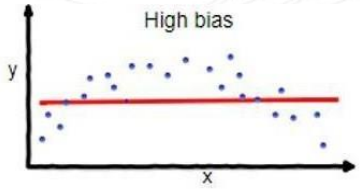
- W przypadku Random Forest podstawowym modelem jest drzewo decyzyjne.



źródło: https://predictivesolutions.pl/jak-udoskonalic-algorytm-drzew-decyzyjnych

Bias-variance tradeoff

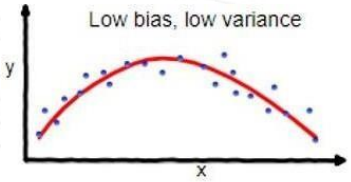
Bias-variance tradeoff



underfitting

Model nie jest w stanie wychwycić skomplikowanych zależności pomiędzy featurami, a zmienną odpowiedzi (target).

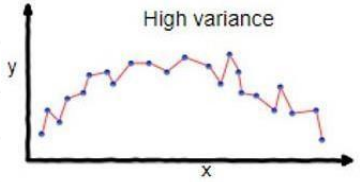
Duże niedopasowanie na zbiorze treningowym i testowym.



Good balance

Model odwzorowuje najważniejsze cechy występujące w danych, ale nie dopasowuje się do szumu w danych.

Dobre dopasowanie na zbiorze testowym oraz na zbiorze treningowym.



overfitting

Model dopasowuje się do szumu losowego występującego w danych treningowych (random noise).

Duże niedopasowanie na zbiorze testowym, pomimo dobrego dopasowania na zbiorze treningowym.

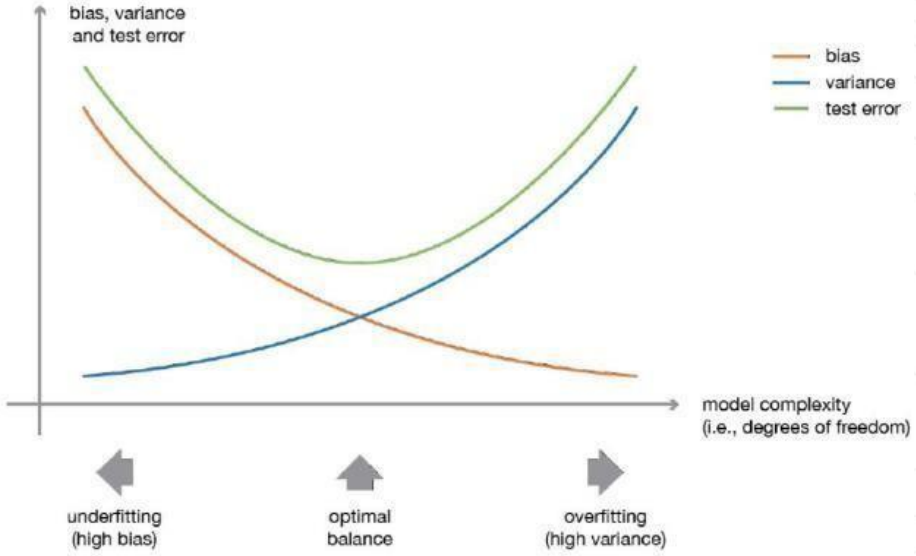
Bias-variance tradeoff - intuicja

Możemy wyobrazić sobie wirtualny “suwak” którym decydujemy jak bardzo model będzie skomplikowany.

W miarę zwiększania poziomu skomplikowania modelu spada “bias error”, a rośnie “variance error”.

Problem polega na znalezieniu idealnego kompromisu pomiędzy tymi wartościami.

Skomplikowane drzewa decyzyjne dobrze dopasowują się do danych treningowych, ale bez regularyzacji mają problem z przeuczaniem się. Lasy decyzyjne mają na celu obniżenie wariancji.



Bootstrap

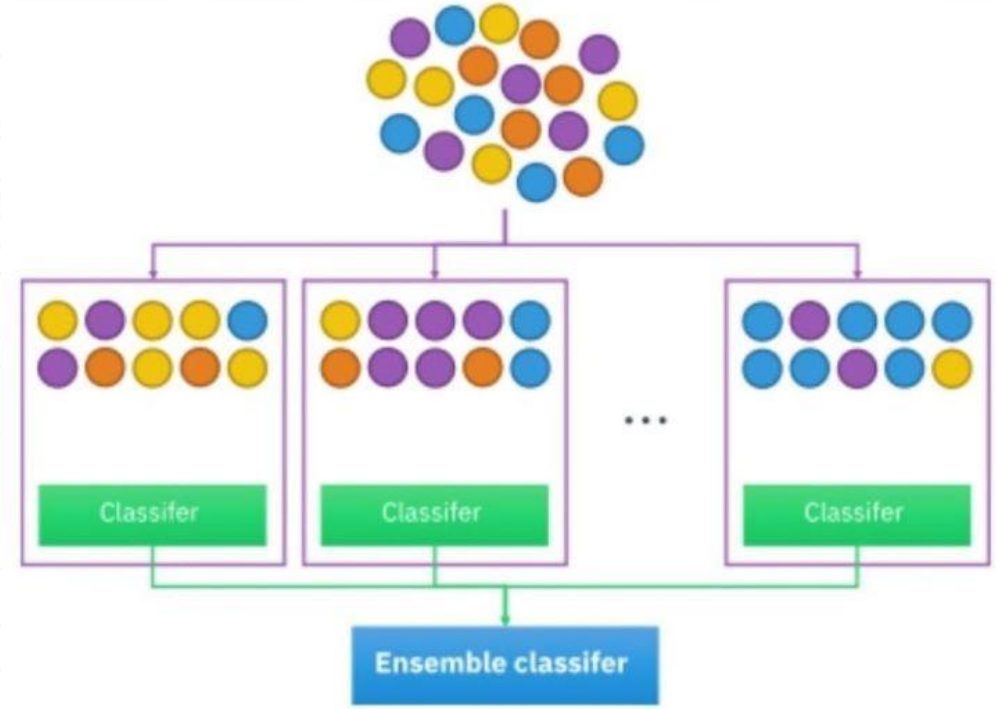
Bagging – Bootstrap aggregating

Bagging to sposób na zmniejszenie variance error.

Zamiast uczyć jedno skomplikowane drzewo uczymy ich wiele wykorzystując technikę bootstrap.

Bootstrap polega na tym, że zamiast uczyć drzewo na danych treningowych uczymy je na tzw. bootstrap sample, czyli zestawie danych stworzonych przez losowanie ze zwracaniem z danych treningowych.

Dlaczego bootstrapujemy?



Ze względu na różne powtórzenia próbek w zbiorze treningowym oraz pominięcie innych próbek w każdej paczce danych powstałe modele będą skupiały się na różnych aspektach.

- Wartości miar nieczystości zbioru danych będą inne.
- Korzenie drzewa będą dzielić dataset od innych zmiennych.
- Niektóre problematyczne gałęzie będą nieobecne w niektórych drzewach.

Parametry modelu i ich optymalizacja

Jak dobrać parametry?

n_estimators - liczba drzew w lesie (większy lepszy, ale dłużej zajmie obliczenie, po krytycznej liczbie drzw wyniki przestają się poprawiać)
max_features - rozmiar losowych podzbiorów cech, które należy wziąć pod uwagę podczas dzielenia węzła (im niższy, tym większa redukcja wariancji, ale także większy wzrost błędu systematycznego)
Empiryczne dobre wartości domyślne to **max_features=None** (zawsze uwzględnianie wszystkich cech zamiast losowego podzbioru) dla problemów regresji i **max_features="sqrt"** dla zadań klasyfikacyjnych.
Dobre wyniki są często osiągane podczas ustawiania **max_depth=None** w połączeniu z **min_samples_split=2** (tj. przy pełnym rozwoju drzew).
Wartości te zwykle nie są optymalne i mogą skutkować modelami zużywającymi dużo pamięci RAM. Najlepsze wartości parametrów powinny zawsze podlegać walidacji krzyżowej. W lasach losowych bootstrap samples są używane domyślnie (**bootstrap=True**).
Podczas korzystania z próbkowania bootstrap jakość modelu można oszacować na próbkach pominiętych. (**oob_score=True** - https://towardsdatascience.com/what-is-out-of-bag-oob-score-in-random-forest-a7fa23d710)
Mamy możliwość zrównoleglenia obliczeń (**n_jobs=-1**).

Jak dobrać parametry?

Bad news

Żeby przekonać się o tym, które wartości parametrów będą odpowiednie musimy zbudować modele i sprawdzić ich jakość, korzystając ze zbioru walidacyjnego, a najlepiej stosując walidację krzyżową (cross validation).

Good news

Python oferuje zestaw narzędzi ułatwiających znalezienie odpowiednich parametrów modelu. Każdy parametr podany podczas konstruowania estymatora (hiperparametr) może być zoptymalizowany w ten sposób.

Podsumowanie

Zalety

- efektywna metoda
- odpowiednia dla dużych zbiorów
- daje oszacowanie, które zmienne są ważne
- odpowiednia dla problemów klasyfikacji i regresji

Wady

- potrzebuje większych zasobów
- proces decyzyjny bardziej skomplikowany niż w przypadku pojedynczego drzewa - trudniejsze do wytłumaczenia
- trudniejsze do wizualizacji od pojedynczego drzewa

Do poczytania w wolnym czasie

1. „Statystyczne systemy uczące się” – J. Koronacki, J. Ćwik
2. Random forests - classification description (berkeley.edu)
3. https://builtin.com/data-science/random-forest-algorithm
4. https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning
5. https://medium.com/@taplapinger/tuning-a-random-forest-classifier-1b252d1dde92

infoShare ACADEMY

Dzięki!