Introduction to ML with Scikit-learn

your first pipeline

confitura

Organizator

O mnie

Waldemar Kołodziejczyk

Zawodowo:

Artificial Intelligence Developer (Atende Software, besmart.energy)
Trener (Sages, kursy ML/AI)

Background akademicki:

Informatyka, **Energetyka** (Politechnika Warszawska)

Obszar zainteresowań:

Machine Learning, Reinforcement Learning, **algorytmika, statystyka**

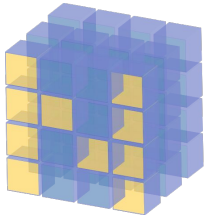
Jak najlepiej wykorzystać ten kurs?

- Materiały są przygotowane tak, aby stanowiły dobry materiał do powtórek, ale istotnym elementem szkolenia jest też część “opowiadana”, więc zachęcam do robienia własnych notatek i komentarzy w kodzie.

- W czasie trwania szkolenia uwzględniony jest też czas na pytania, więc zachęcam do ich zadawania. ;)

- Slajdy są cały rozwijane i staram się ze szkolenia na szkolenie je usprawniać, uzupełniać i aktualizować.

Stack technologiczny



NumPy

pandas

$$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$$

ANACONDA®

Agenda

1. Wstęp, nazewnictwo i źródła danych
2. Preprocessing danych
3. Problem klasyfikacji
4. Miary oceny modeli przy problemie klasyfikacji
5. Podsumowanie i zakończenie

Machine Learning z lotu ptaka

Machine Learning

Supervised Learning

Regresja
- predictions
- process optimization

Klasyfikacja
- image classification
- diagnostics
- fraud detection

Unsupervised Learning

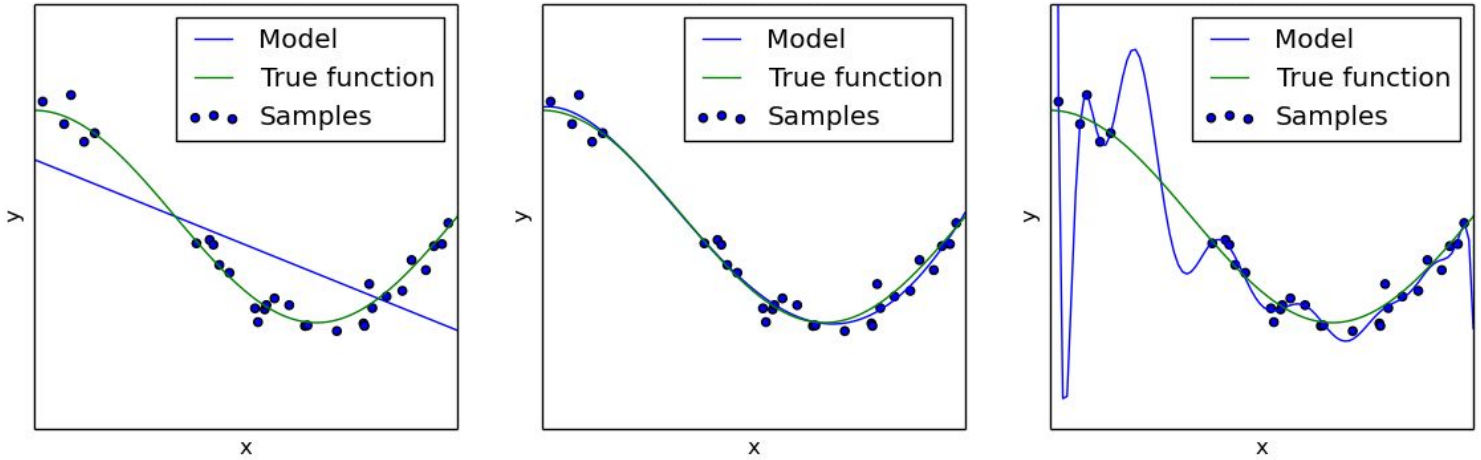
Klastrowanie
- recommendation system
- targeted marketing
- customer segmentation

Redukcja wymiarowości
- Big Data visualisation
- feature elicitation
- data compression

Reinforcement Learning

- games
- real-time decisions
- robot / dron navigation

Bias i Wariancja

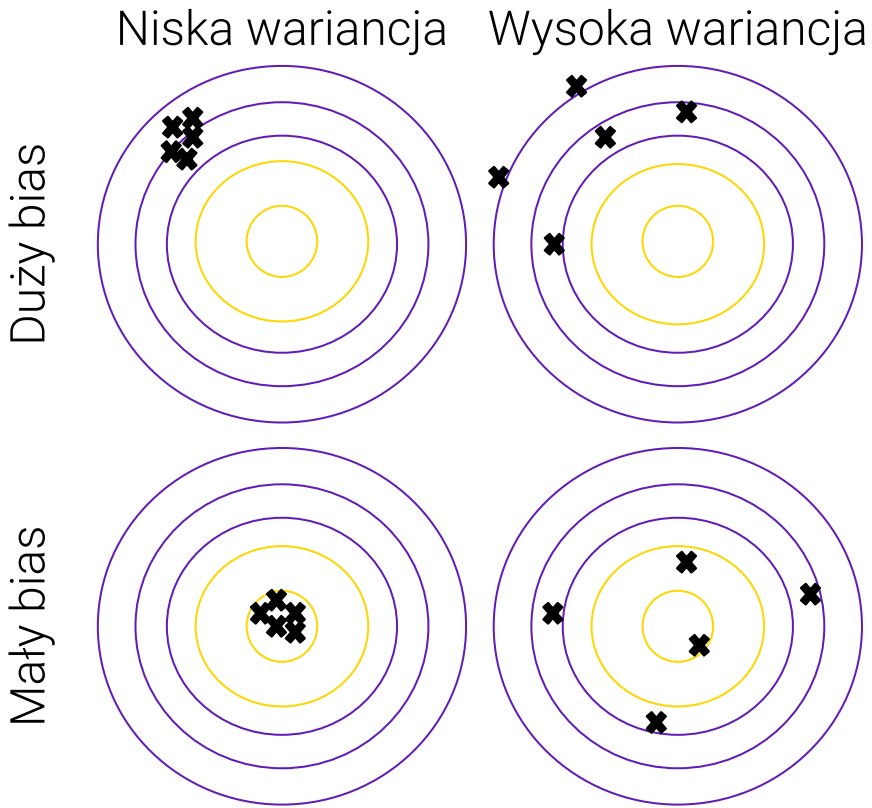


scikit-learn.org

Bias i Wariancja

Bias (obciążenie) - niemożność uchwycenia przez model prawdziwej zależności pomiędzy wejściem a wyjściem.

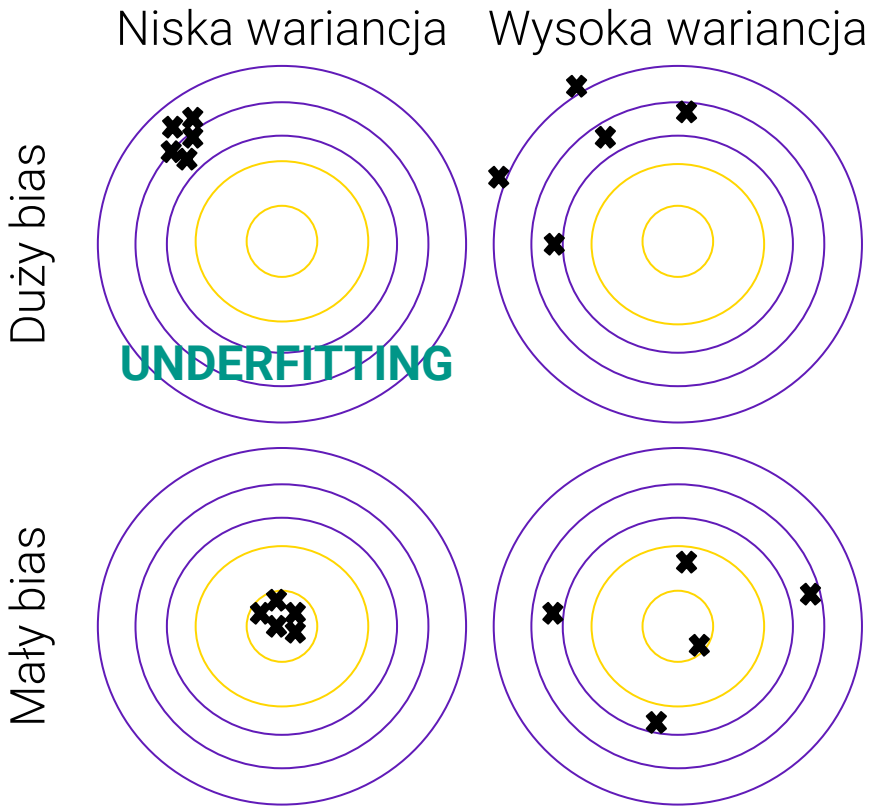
Wariancja - błąd wynikający z wrażliwości na małe zmiany danych treningowych (np. model ”uczy się” szumu w danych)



Bias i Wariancja

Sposoby na underfitting:

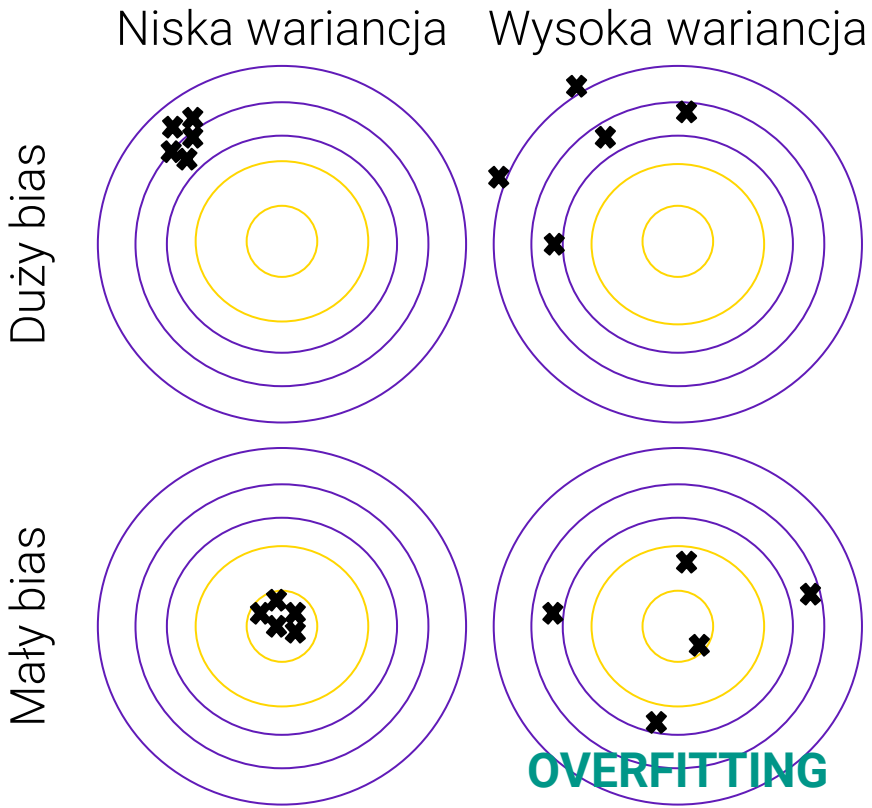
- dobór bardziej złożonego modelu z większą ilością parametrów,
- zwiększenie liczby feature'ów do modelu (ang. feature engineering),
- zmniejszenie ograniczeń modelu (np. usunięcie parametru regularyzacji).



Bias i Wariancja

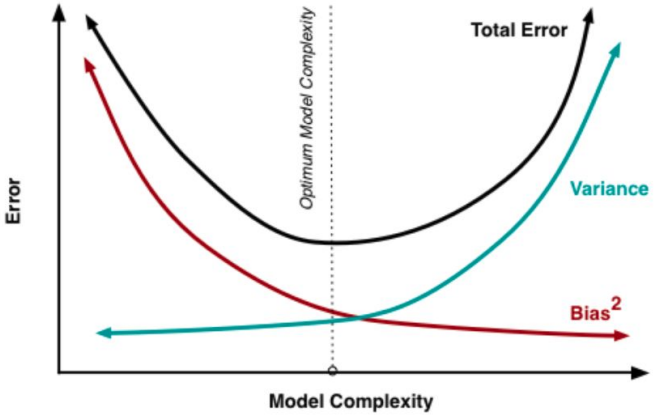
Sposoby na overfitting:

- pozyskanie większej ilości danych uczących,
- uproszczenie modelu lub zmiana na mniej złożony lub zawierający mniej parametrów,
- zmniejszenie zaszumienia danych, poprzez np. usunięcie błędnych danych lub outlierów.

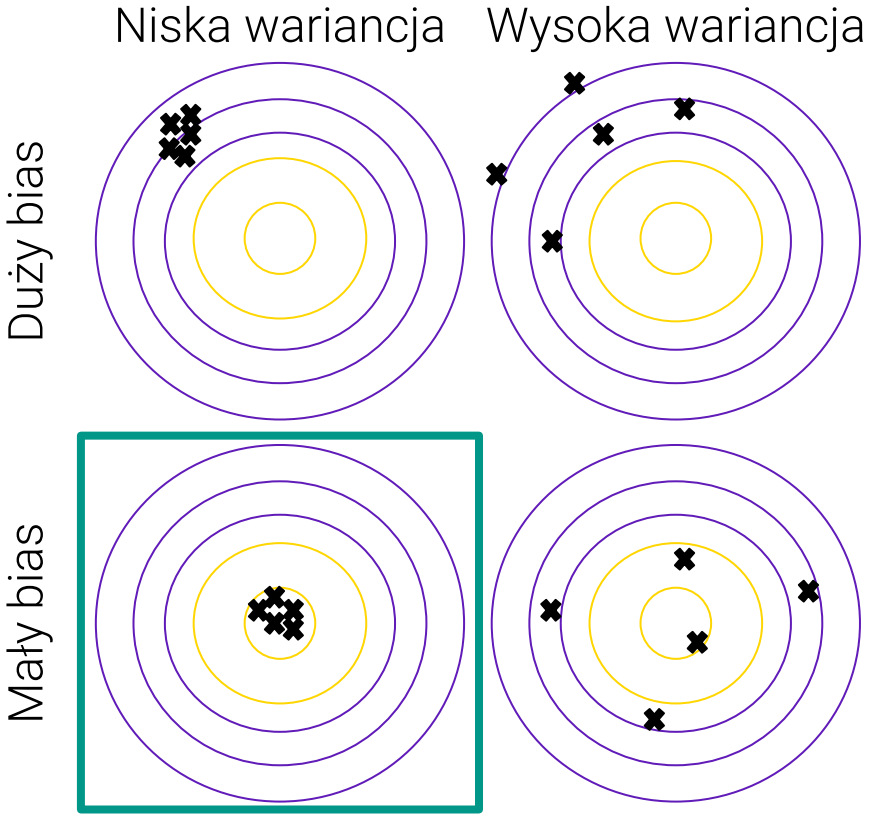


Bias i Wariancja

Model zbalansowany - wynik kompromisu pomiędzy biasem a wariancją



scott.fortmann-roe.com



Preprocessing danych

Kluczowe pojęcia:

1. Rodzaje zmiennych
2. Wyszukiwanie wartości odstających
3. Uzupełnianie brakujących danych
4. Encoding zmiennych kategorycznych
5. Skalowanie danych
6. Podział zbioru danych na treningowe, walidacyjne i testowe
7. Walidacja krzyżowa (klasyczna i dla szeregów czasowych)

Rodzaje zmiennych

Numeryczne	Kategoryczne	Szeregi czasowe	Złożone
Ciągłe	Binarne	Numeryczne	Tekst
Dyskretne	Nominalne	Kategoryczne	Obrazy
	Uszeregowane	Mieszane	Video
			Audio

Rodzaje zmiennych

Numeryczne	**Kategoryczne**	Szeregi czasowe	Złożone
Ciągłe	Binarne	Numeryczne	Tekst
Dyskretne	Nominalne	Kategoryczne	Obrazy
	Uszeregowane	Mieszane	Video
			Audio

Rodzaje zmiennych

Numeryczne	Kategoryczne	**Szeregi czasowe**	Złożone
Ciągłe	Binarne	Numeryczne	Tekst
Dyskretne	Nominalne	Kategoryczne	Obrazy
	Uszeregowane	Mieszane	Video
			Audio

Rodzaje zmiennych

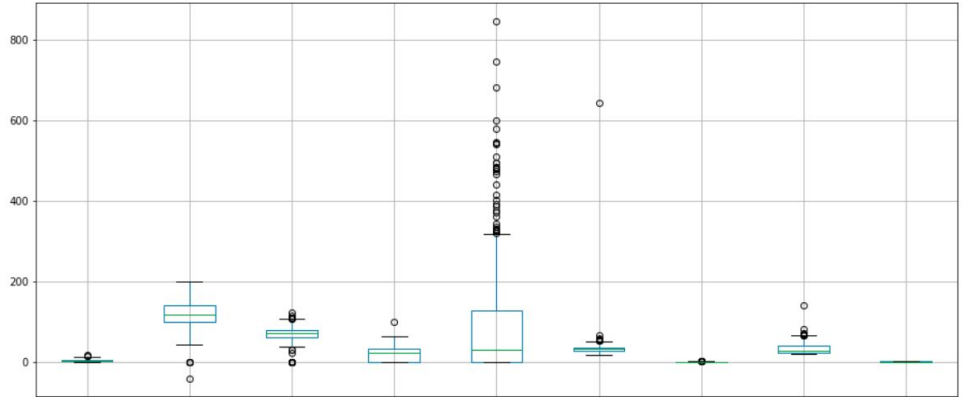
Numeryczne	Kategoryczne	Szeregi czasowe	**Złożone**
Ciągłe	Binarne	Numeryczne	Tekst
Dyskretne	Nominalne	Kategoryczne	Obrazy
	Uszeregowane	Mieszane	Video
			Audio

Rodzaje zmiennych

Numeryczne	**Kategoryczne**	**Szeregi czasowe**	**Złożone**
Ciągłe	Binarne	Numeryczne	Tekst
Dyskretne	Nominalne	Kategoryczne	Obrazy
	Uszeregowane	Mieszane	Video
			Audio

Jak odnaleźć wartości odstające (ang. outliery)

1. **Na podstawie wiedzy ‘eksperckiej’ -** jako doświadczeni w danej dziedzinie możemy z góry być pewni niektórych ograniczeń naszych zmiennych, np. zajmowana pamięć czy wiek człowieka nie mogą być ujemne
2. **Na podstawie rozkładów statystycznych -** Wizualizując rozkłady można wyodrębnić pojedyncze próbki o wartościach zupełnie innych niż reszta wartości znajdujących się w danej dystrybucji, szczególnie przydatne są w takiej sytuacji box-ploty



Typy brakujących danych:

1. **Missing at Random** (MAR) - przyczyny brakujących danych nie korelują z typem obserwacji, ich obecność w konkretnych miejscach nie wnosi dodatkowej informacji, na przykład błędy przy odczycie z bazy danych

2. **Missing not at Random** (MNAR) - brakujące dane nie są przypadkowo rozmieszczone w zbiorze danych i mogą wnosić dodatkową informację, na przykład osoby z wyższymi zarobkami rzadziej będą się dzielić wysokością swojej pensji, pozostawiając puste pole w ankiecie

Brakujące dane

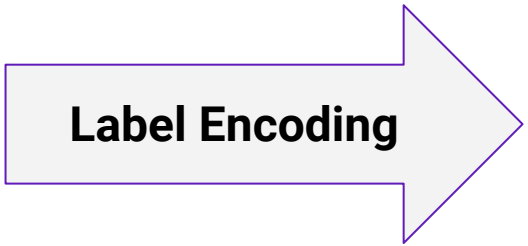
Imputacja	**Usuwanie**
mediana, średnia	poszczególne rekordy (wiersze)
wartość losowa	całe feature'y (kolumny)
estymacja złożona (np. interpolacja)	
utworzenie feature'a 'niewiadomej'	

Brakujące dane

Imputacja	Usuwanie
mediana, średnia	poszczególne rekordy (wiersze)
wartość losowa	całe feature'y (kolumny)
estymacja złożona (np. interpolacja)	
utworzenie feature'a 'niewiadomej'	

Encoding zmiennych kategorycznych

height	name
156	‘Ted’
187	‘Amy’
124	‘Max’
184	‘Max’



height	name
156	0
187	1
124	2
184	2

! czasem nieodpowiednie

Encoding zmiennych kategorycznych

height	name
156	‘Ted’
187	‘Amy’
124	‘Max’
184	‘Max’

height	name_Ted	name_Amy	name_Max
156	1	0	0
187	0	1	0
124	0	0	1
184	0	0	1

? czy da się uprościć?

Encoding zmiennych kategorycznych

height	name
156	‘Ted’
187	‘Amy’
124	‘Max’
184	‘Max’

One-Hot Encoding

height	name_Ted	name_Amy	name_Max
156	1	0	0
187	0	1	0
124	0	0	1
184	0	0	1

Wypróbujmy to!

Kiedy skalować dane?

Zawsze, gdy algorytm operuje na odległościach lub gradientach.

Algorytmy szczególnie wrażliwe na nieprzeskalowane feature'y, to np.:

- Metoda Najbliższych Sąsiadów,
- modele liniowe i neuronowe,
- Maszyny Wektorów Wspierających,
- Principal Component Analysis.

Kiedy skalować dane?

Zawsze, gdy algorytm operuje na odległościach lub gradientach.

Niektóre algorytmy są niewrażliwe na nieprzeskalowane feature'y, np.:

- modele oparte o drzewa decyzyjne,
- modele wykorzystujące statystykę Bayes'owską.

Skalowanie danych

1. **Standaryzacja** (x'.mean() = 0 , x'.std() = 1)

   x' = (x - x.mean()) / x.std()

2. **Normalizacja średniej** (x'.mean() = 0 , -1 < x' < 1)

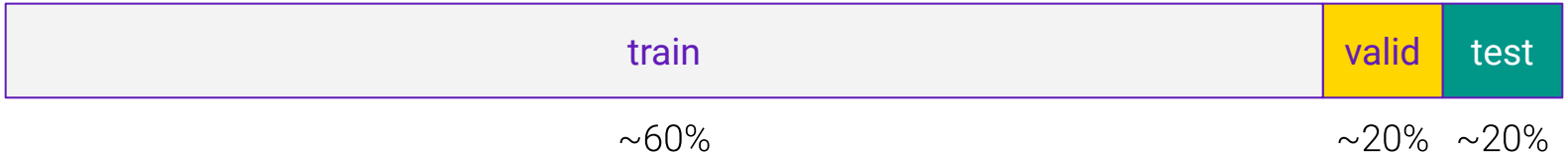
   x' = (x - x.mean()) / (x.max() - x.min())

3. **Skalowanie Min-Max** (0 < x' < 1)

   x' = (x - x.min()) / (x.max() - x.min())

Wypróbujmy to!

Podział zbioru treningowego



train - zbiór treningowy, na którym uczymy model

valid - zbiór walidacyjny, na którym testujemy obecność np. overfittingu

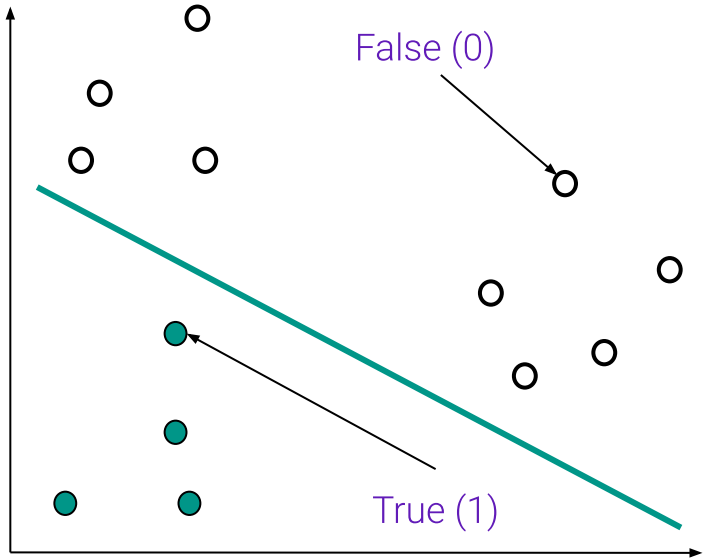
test - zbiór, na którym testujemy nasz ostateczny model przed deploymentem

Problem klasyfikacji

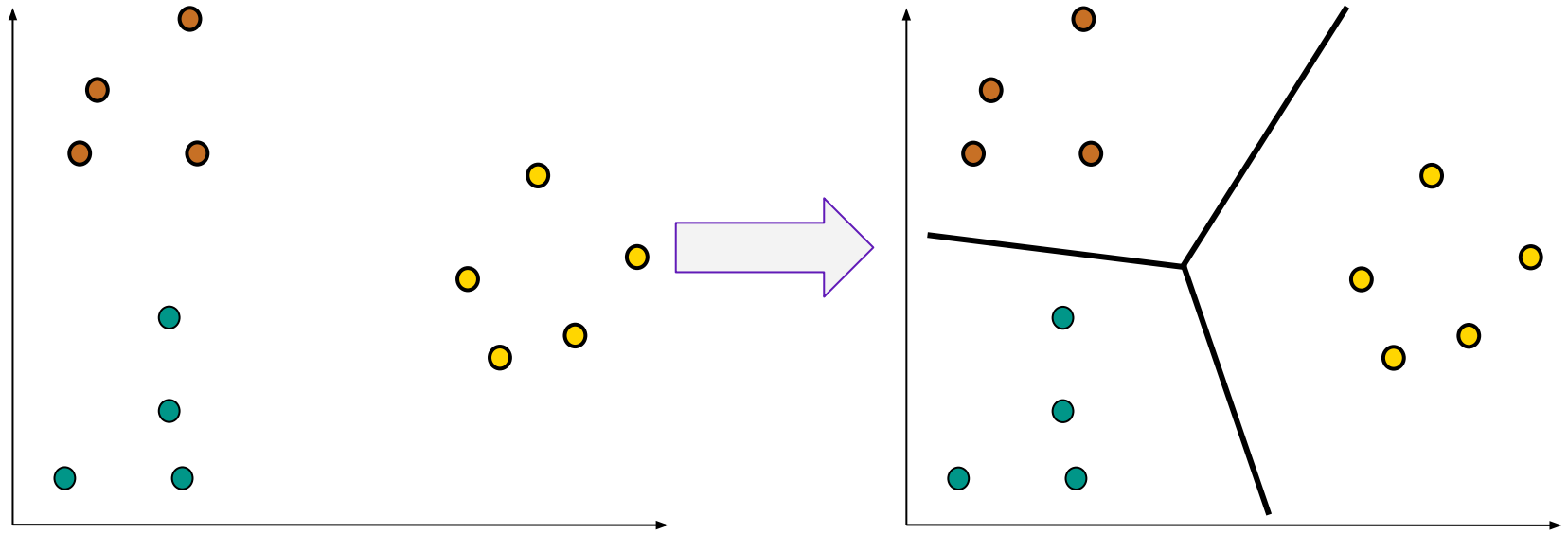
Klasyfikacja binarna (dwuklasowa)

Podstawowa forma klasyfikacji, odpowiada na pytania typu: ‘Czy obiekt należy do klasy X?’ - Tak lub Nie

Do tej postaci sprowadzane są wszystkie problemy klasyfikacji wieloklasowej.

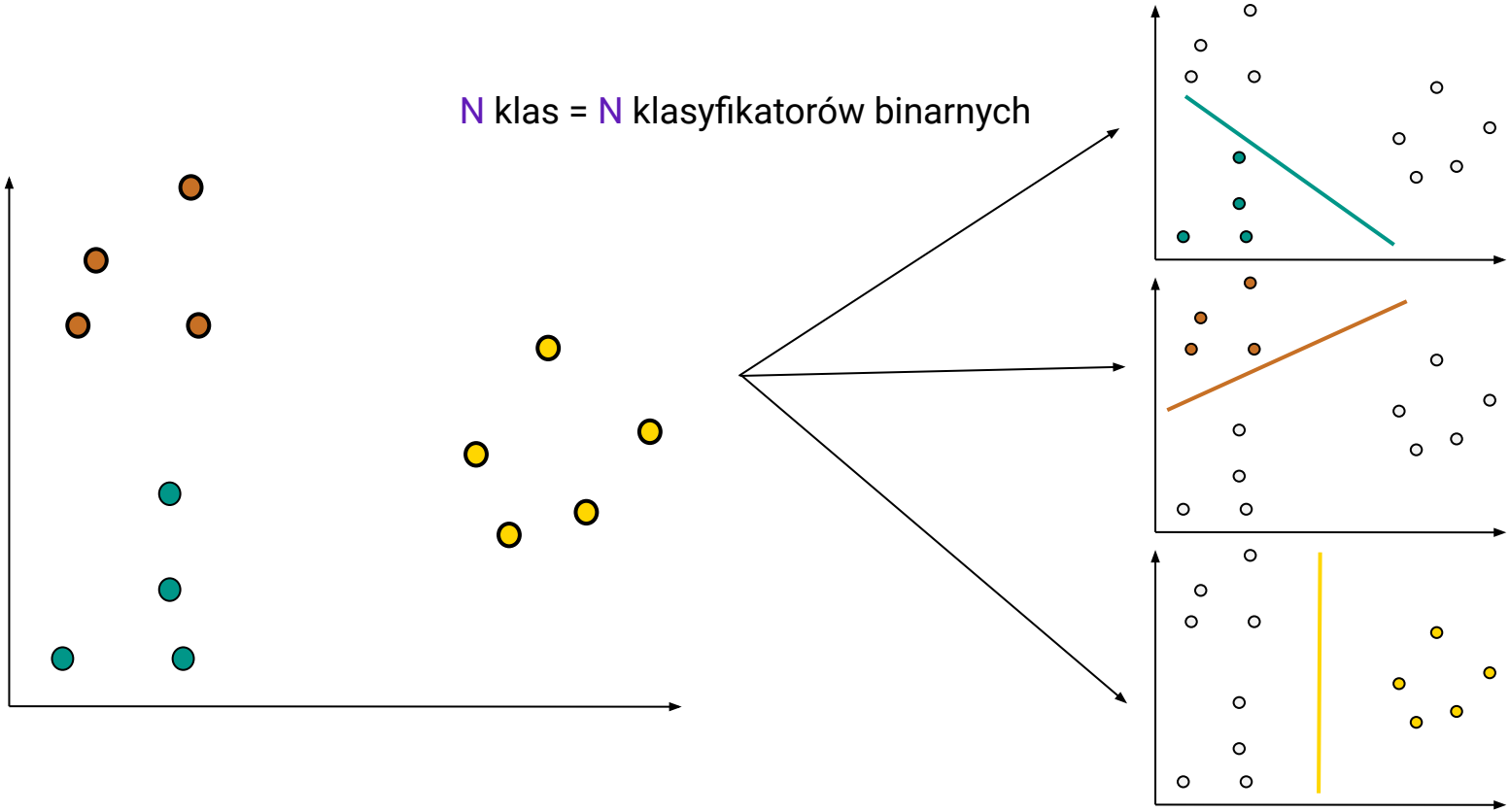


Klasyfikacja wieloklasowa



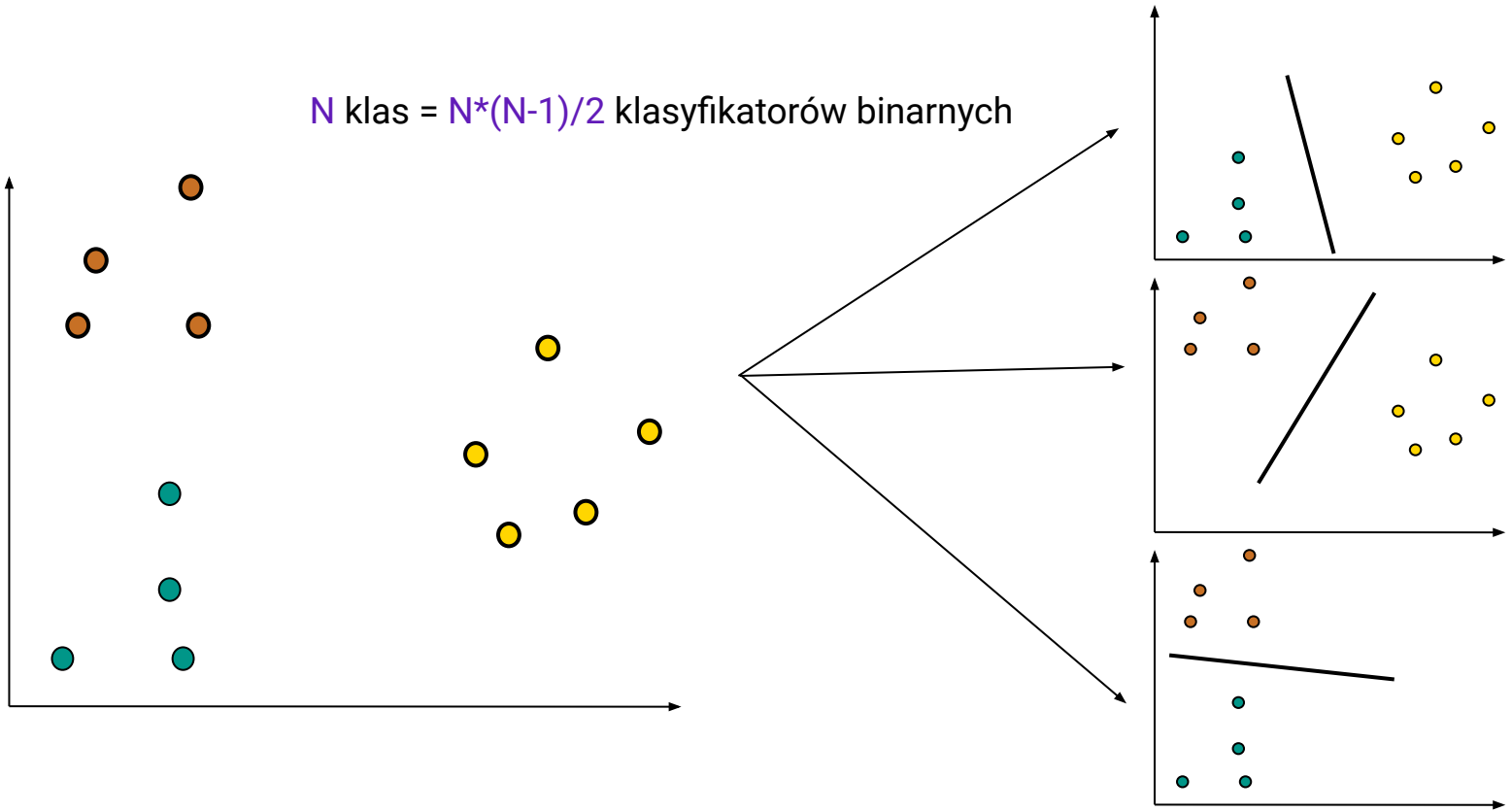
Klasyfikacja wieloklasowa - One vs. All

N klas = N klasyfikatorów binarnych



Klasyfikacja wieloklasowa - All vs. All

N klas = N*(N-1)/2 klasyfikatorów binarnych

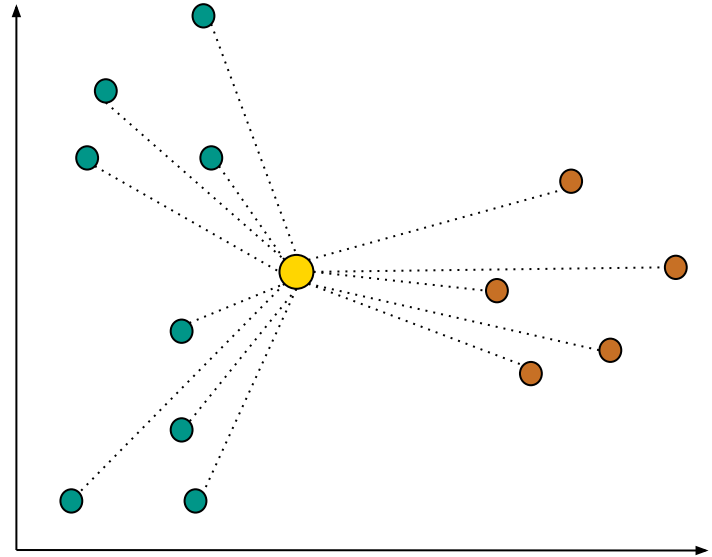


Algorytmy, które zastosujemy, to:

1. Metoda Najbliższych Sąsiadów (K-NN)
2. Regresja Liniowa + Logistyczna
3. Maszyny Wektorów Wspierających (SVM + Kernel)
4. Drzewa Decyzyjne
5. Lasy Drzew Losowych
6. Naiwny Bayes
7. Sieci Neuronowe

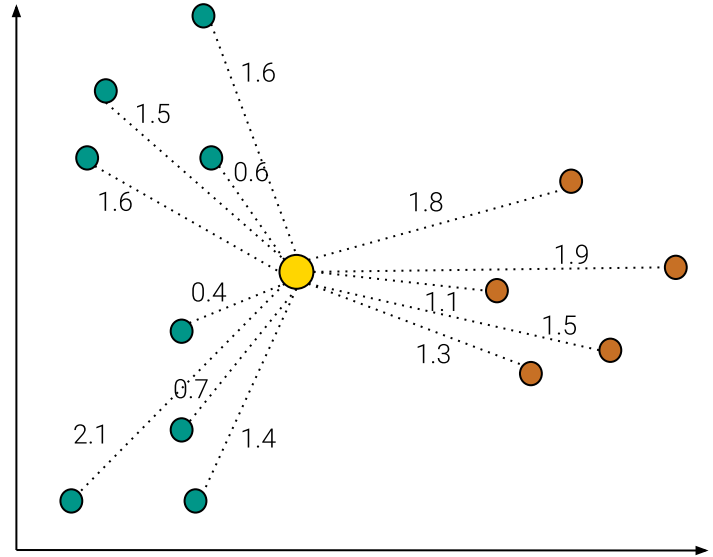
Metoda Najbliższych Sąsiadów (K-NN)

1. Umieść nowy punkt dla przestrzeni cech wraz z innymi oznaczonymi punktami.



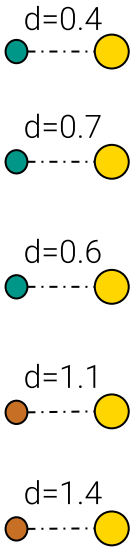
Metoda Najbliższych Sąsiadów (K-NN)

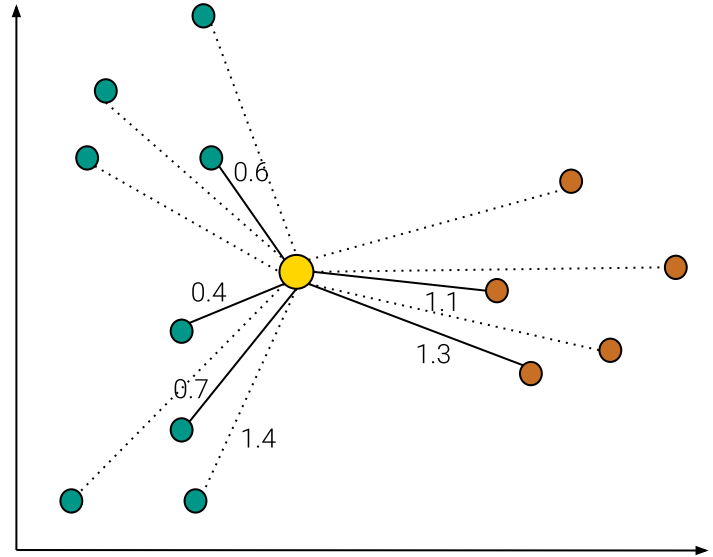
2. Dla każdego punktu w zbiorze trenującym oblicz odległości do nowego punktu.



Metoda Najbliższych Sąsiadów (K-NN)

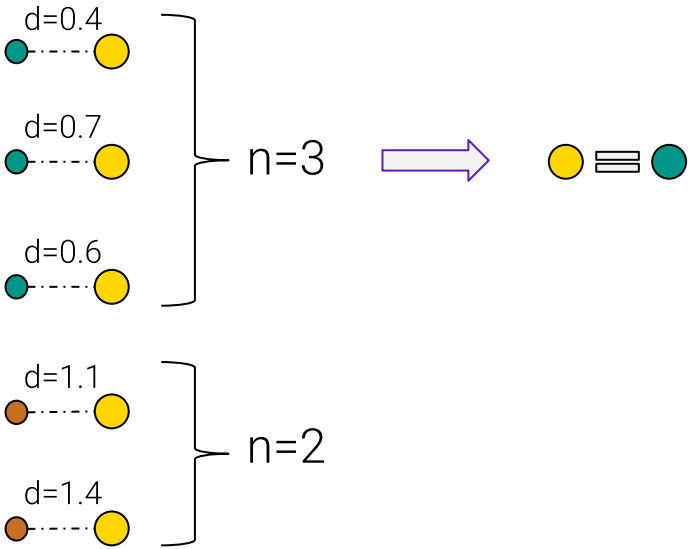
3. Wybierz **K** najmniejszych odległości do nowego punktu.

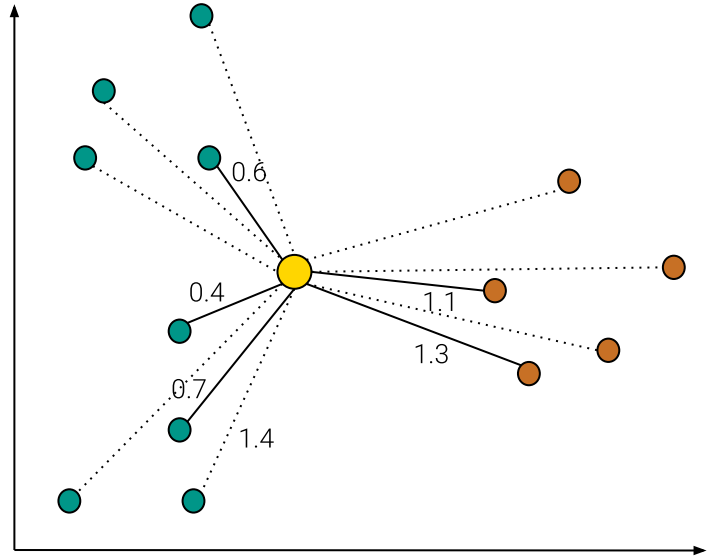




Metoda Najbliższych Sąsiadów (K-NN)

4. Spośród **K** punktów, oblicz liczność każdej z klas i nowemu punktowi przypisz najliczniejszą.

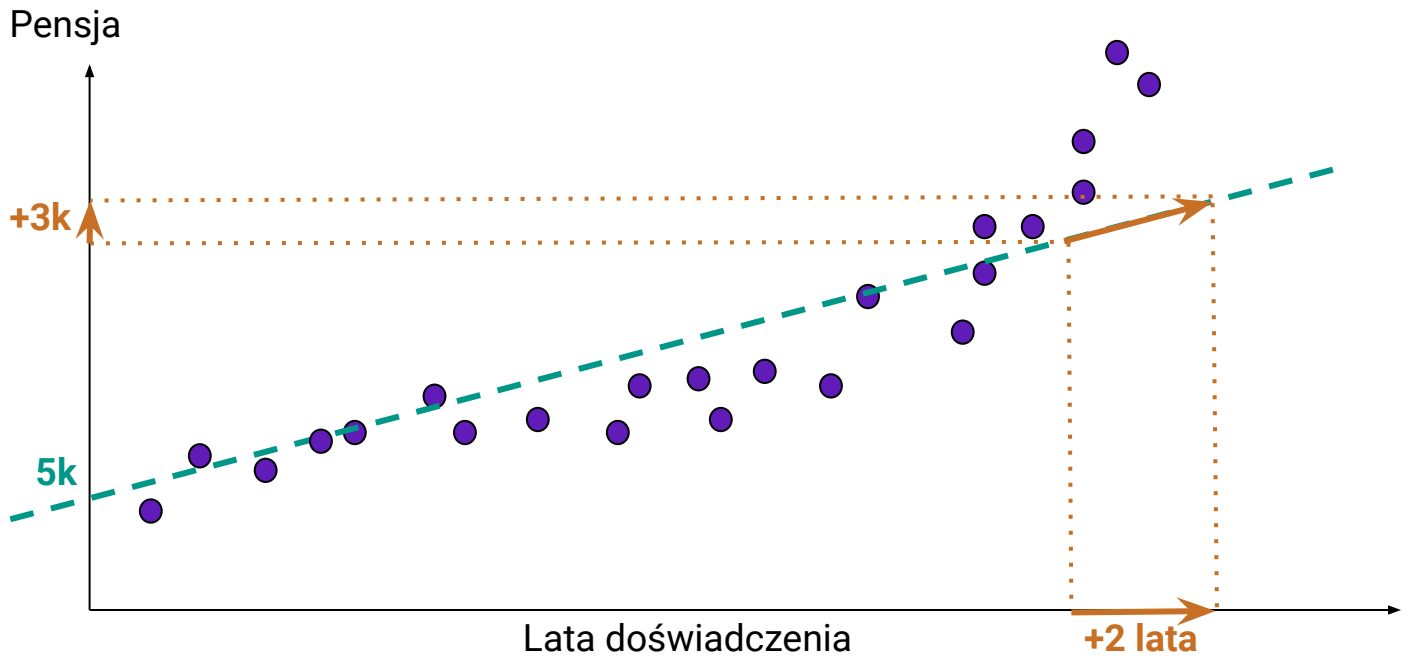




Wypróbujmy to!

Regresja Liniowa

Pensja = 5000 + (3000/2) * Lata doświadczenia



Regresja Liniowa

Pensja = 5000 + (3000/2) * Lata doświadczenia

Formalnie:

$$\hat{y} = \theta_0 + \theta_1 x_1$$

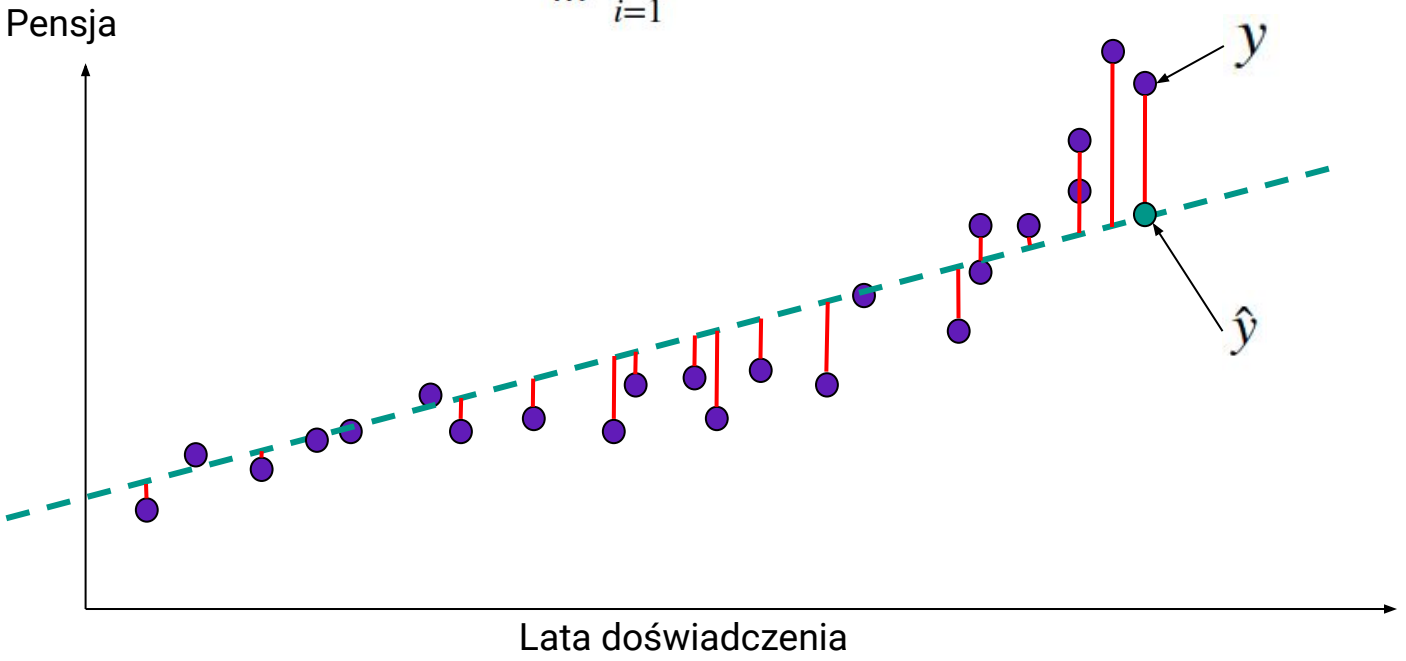
dla jednej zmiennej

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n$$

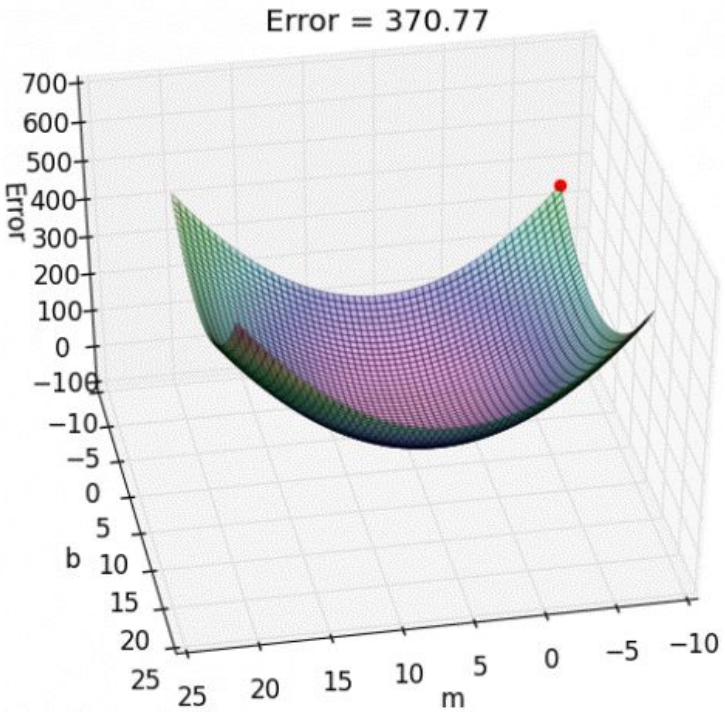
dla n zmiennych

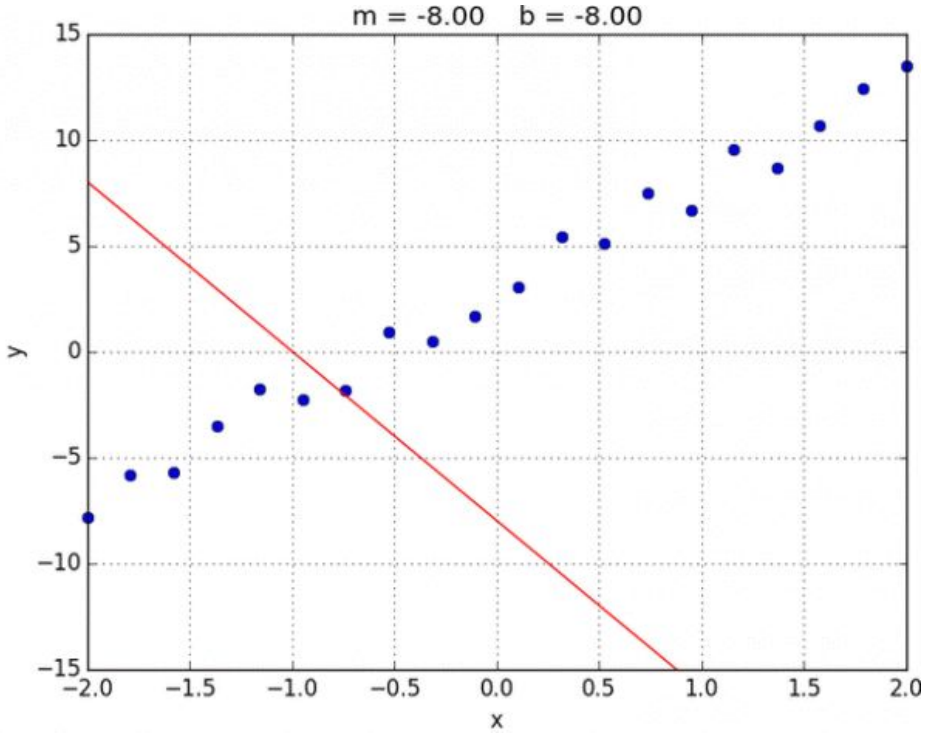
Regresja Liniowa - funkcja kosztu

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$



Regresja Liniowa

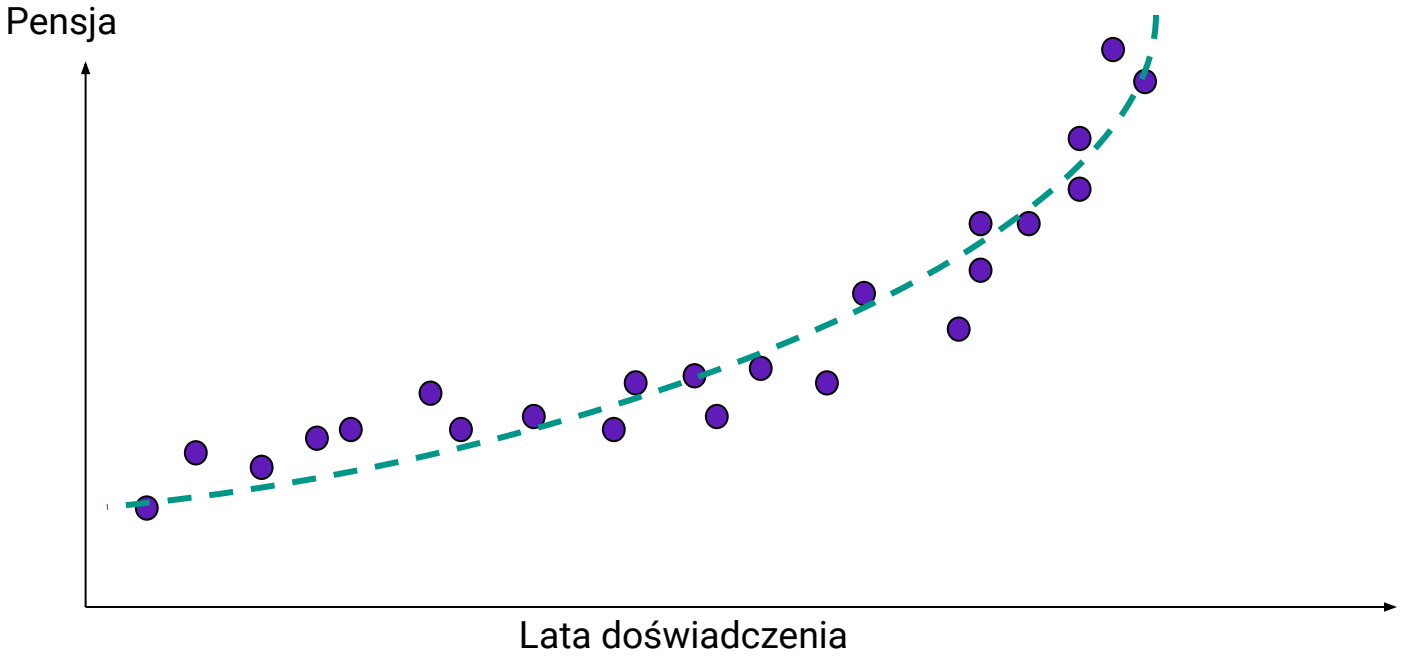




hackernoon.com

Regresja Wielomianowa

Pensja = 5000 + (3000/2) * Lata doświadczenia + 50 * Lata doświadczenia^2



Regresja Wielomianowa

Pensja = 5000 + (3000/2) * Lata doświadczenia + 50 * Lata doświadczenia^2

Formalnie:

$$\hat{y} = \theta_0 + \theta_{11}x_1 + \theta_{12}x_1^2$$

dla **jednej** zmiennej stopnia **2**

$$\hat{y} = \theta_0 + \theta_{11}x_1 + \theta_{12}x_1^2 + \theta_{21}x_2 + \theta_{22}x_2^2 + \ldots + \theta_{n1}x_n + \theta_{n2}x_n^2$$

dla **n** zmiennych stopnia **2**

$$\hat{y} = \theta_0 + \theta_{11}x_1 + \theta_{12}x_1^2 + \theta_{13}x_1^3 + \ldots + \theta_{1p}x_1^p$$

dla **jednej** zmiennej stopnia **p**

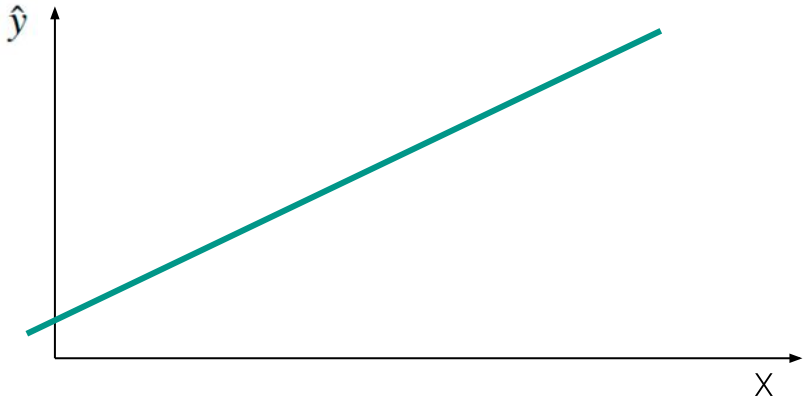
.....

dla **n** zmiennych stopnia **p**

Regresja Logistyczna

Regresja Liniowa

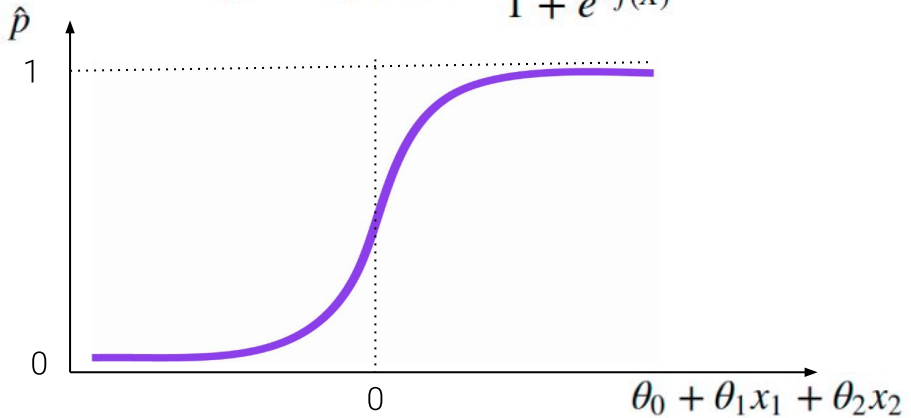
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n$$



Regresja Logistyczna

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\hat{p} = \sigma(f(X)) = \frac{1}{1 + e^{-f(X)}}$$



Regresja Logistyczna

Regresja Liniowa

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n$$

Funkcja kosztu:

$$L(\theta) = MSE = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

Regresja Logistyczna

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\hat{p} = \sigma(f(X)) = \frac{1}{1 + e^{-f(X)}}$$

$$\hat{p} = \sigma(\theta_0 + \theta_1 x_1 + \ldots) = \frac{1}{1 + e^{\theta_0 + \theta_1 x_1 + \ldots}}$$

Funkcja kosztu:

$$L(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^i log(\hat{p}^{(i)}) + (1 - y^i) log(1 - \hat{p}^{(i)}) \right]$$

Wypróbujmy to!

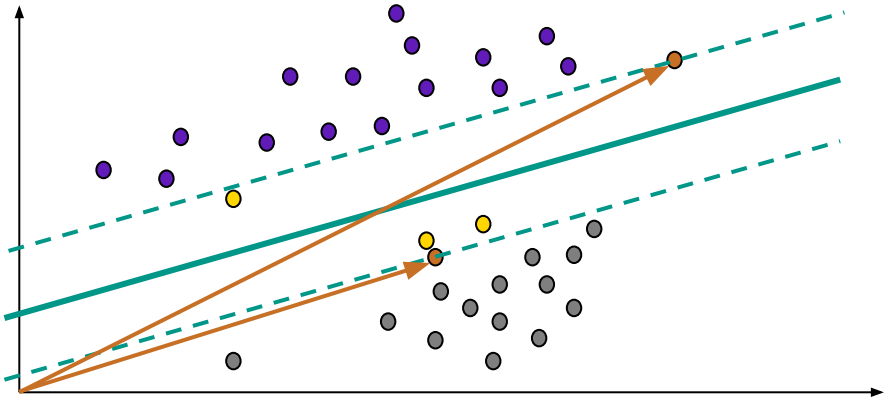
Maszyny Wektorów Wspierających (SVM)

SVM

SVC

Support Vector Classification

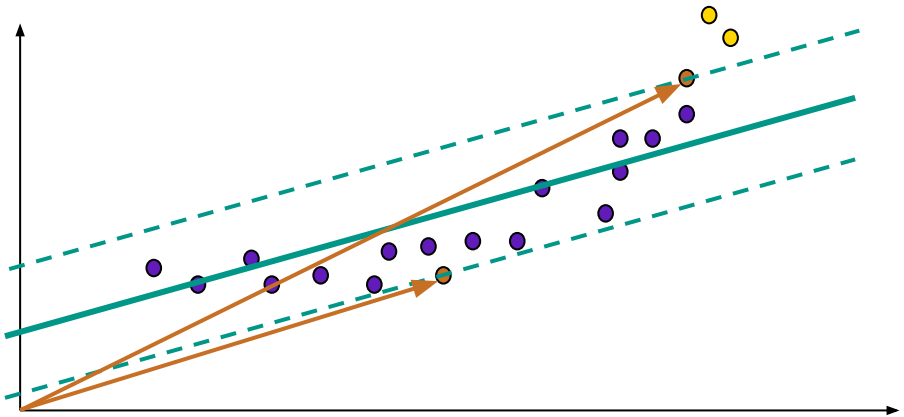
Cel: **maksymalizacja** marginesu



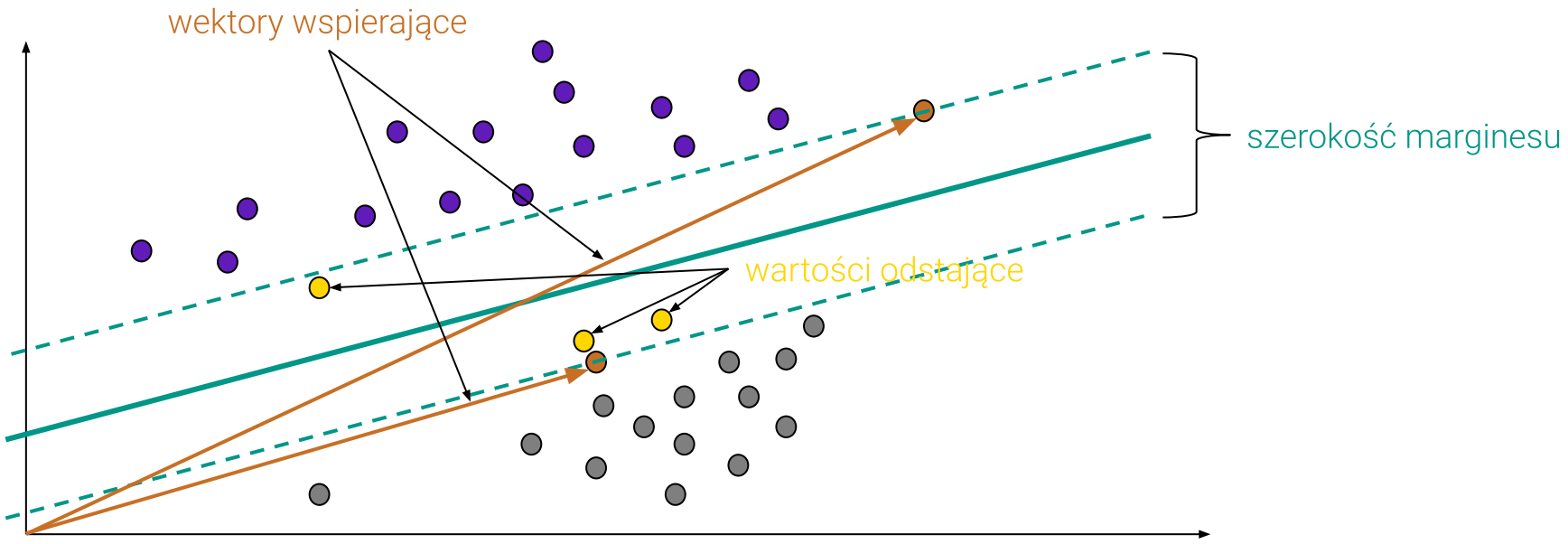
SVR

Support Vector Regression

Cel: **minimalizacja** marginesu



Maszyny Wektorów Wspierających

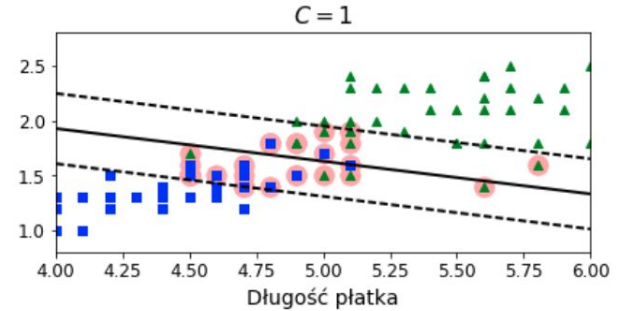


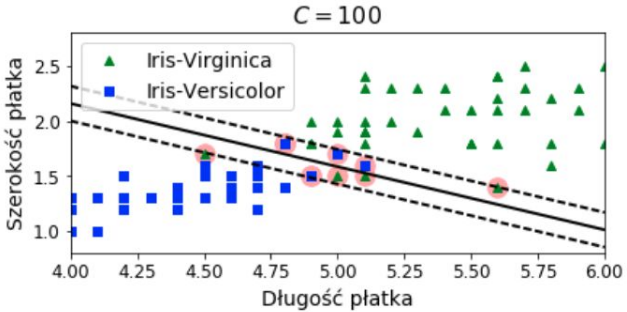
Maszyny Wektorów Wspierających

Hiperparametr C

Określa wagę funkcji kosztu,
umożliwia regularyzację modelu.

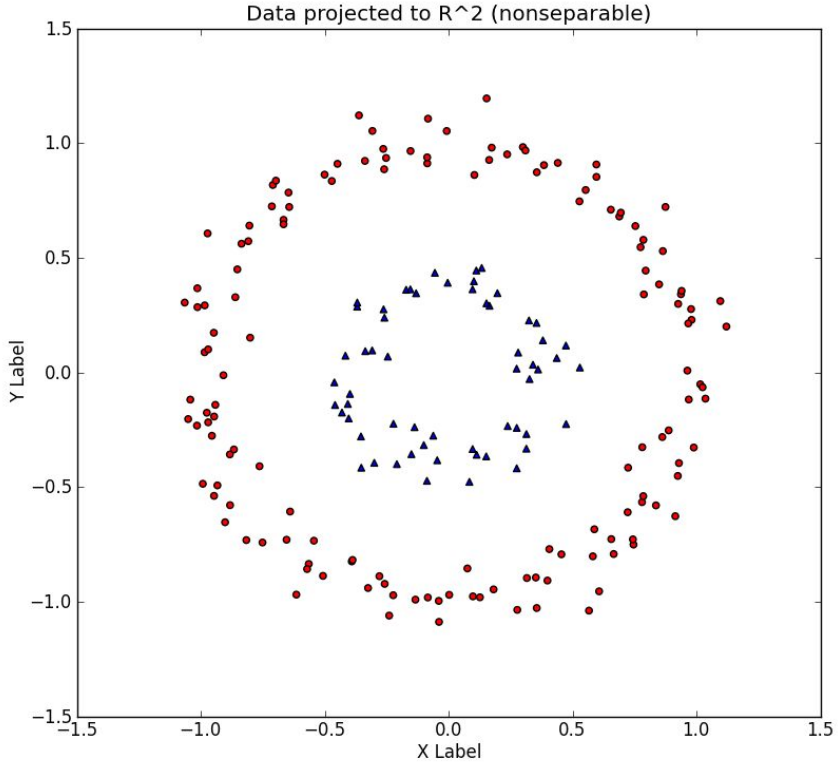
Im większe C, tym węższa ulica.

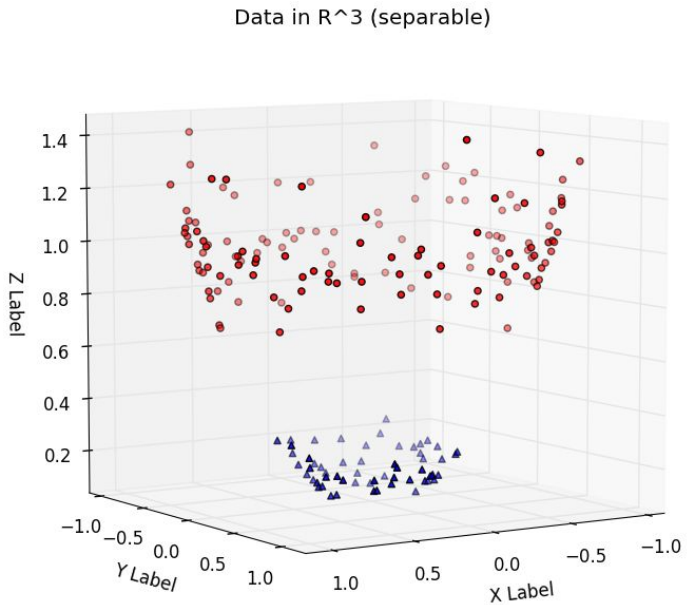




"Uczenie Maszynowe", A. Geron

Maszyny Wektorów Wspierających - Kernele



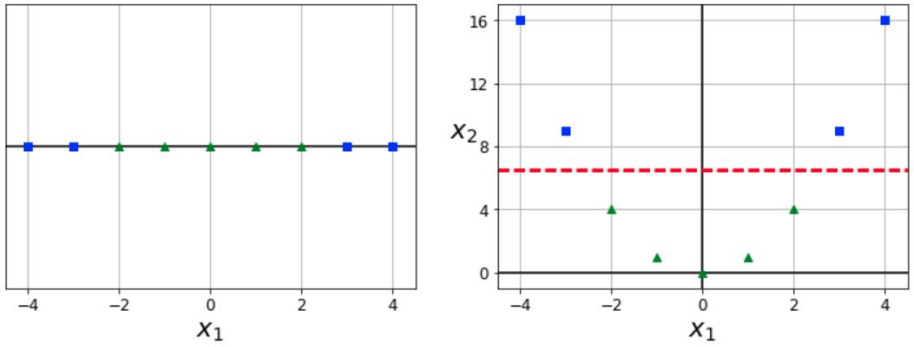


towardsdatascience.com

Maszyny Wektorów Wspierających - Kernele

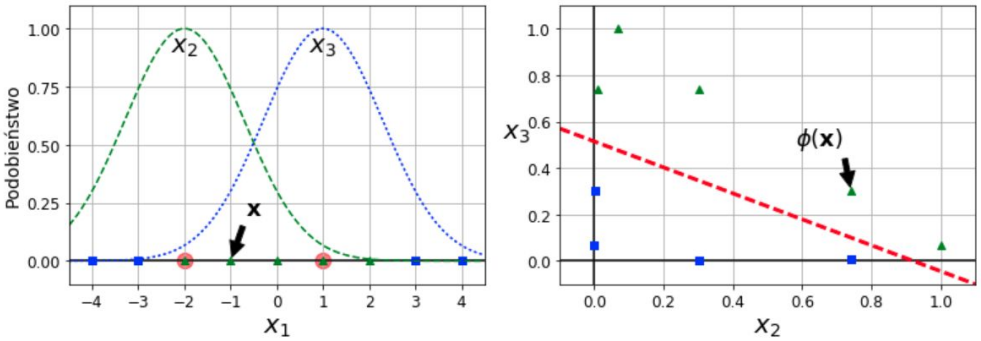
Najpopularniejsze kernele w SVM:

Wielomianowe: np. kwadratowe



Oparte o funkcję podobieństwa:
(ang. Radial Basis Function, RBF)

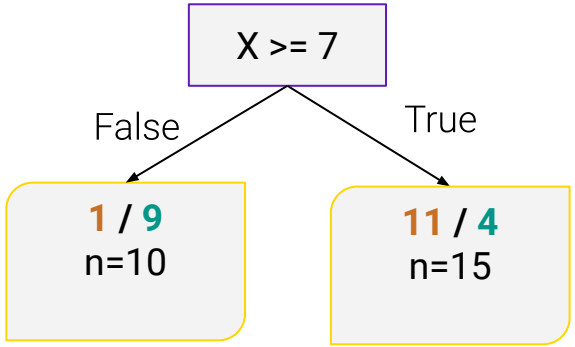
$$\phi(x, l) = exp(-\gamma|x - l|^2), \gamma = \frac{1}{2\sigma^2}$$



"Uczenie Maszynowe", A. Geron

Wypróbujmy to!

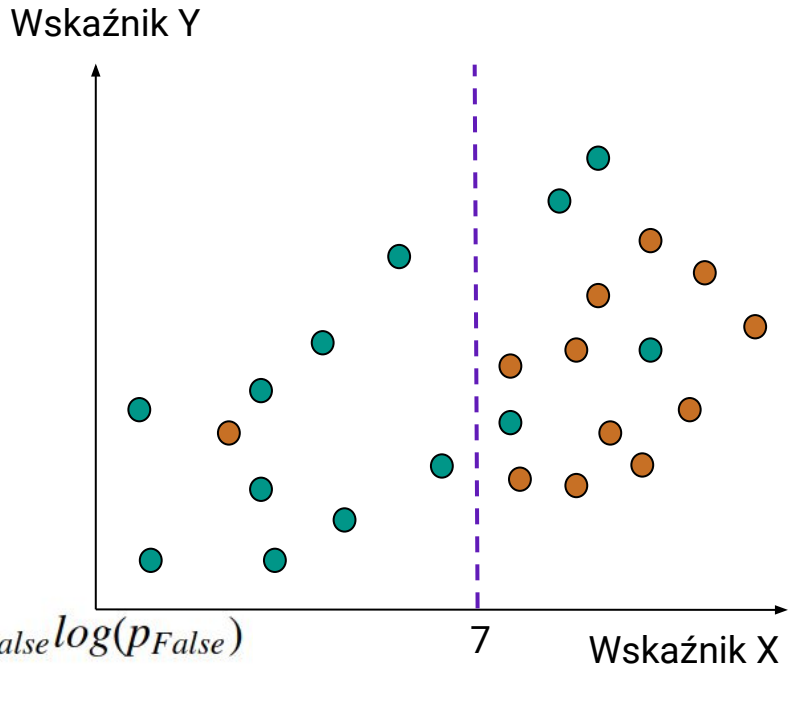
Drzewa decyzyjne



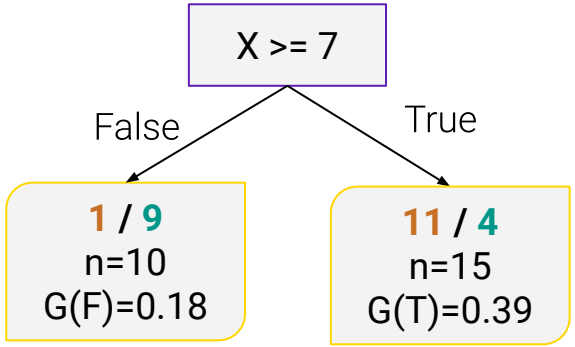
Indeksy Gini i Entropia (ang. impurity indexes):

$$Gini(L) = 1 - \sum_{i=1}^{k} p_i^2 = 1 - p_{True}^2 - p_{False}^2$$

$$Entrophy(L) = -\sum_{i=1}^{k} p_i log(p_i) = -p_{True} log(p_{True}) - p_{False} log(p_{False})$$

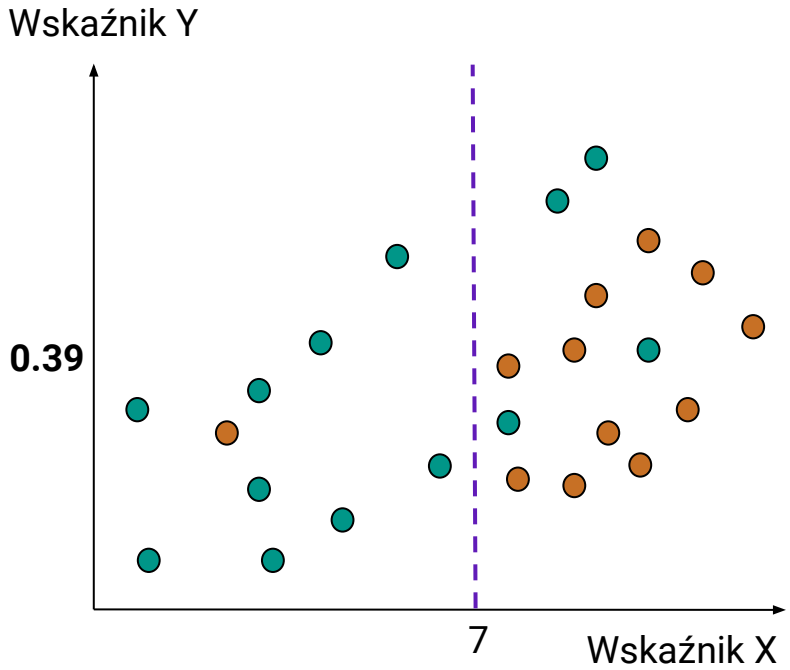


Drzewa decyzyjne



G(F) = 1 - $(1/10)^2$ - $(9/10)^2$ = **0.18**

G(T) = 1 - $(11/15)^2$ - $(4/15)^2$ = **0.39**



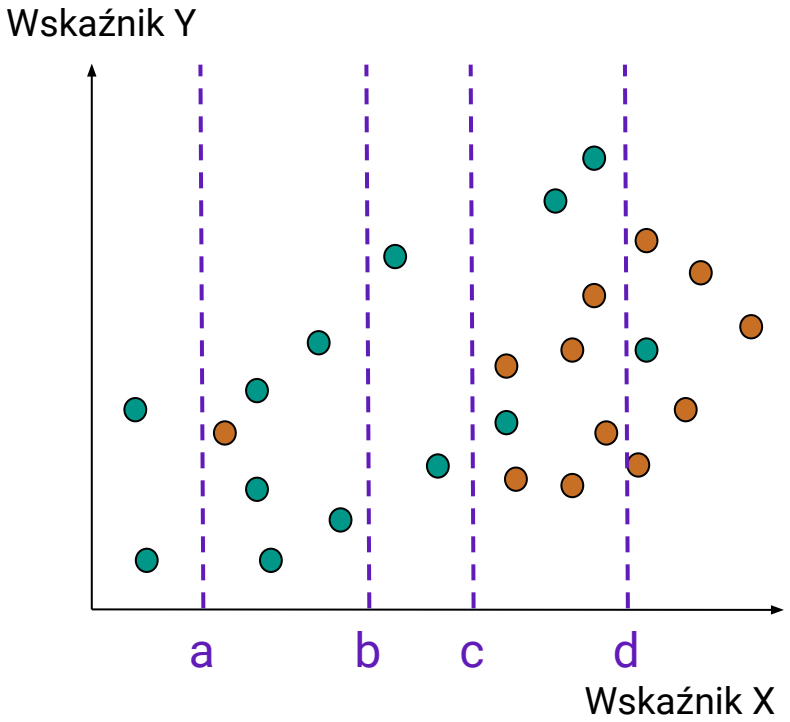
Dla roota kryterium:

G(X>=7) = (10/25) * G(F) + (15/25) * G(T)

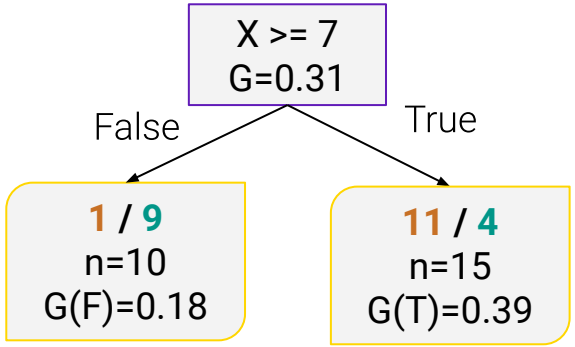
G(X>=7) = (10/25) * 0.18 + (15/25) * 0.39 = **0.31**

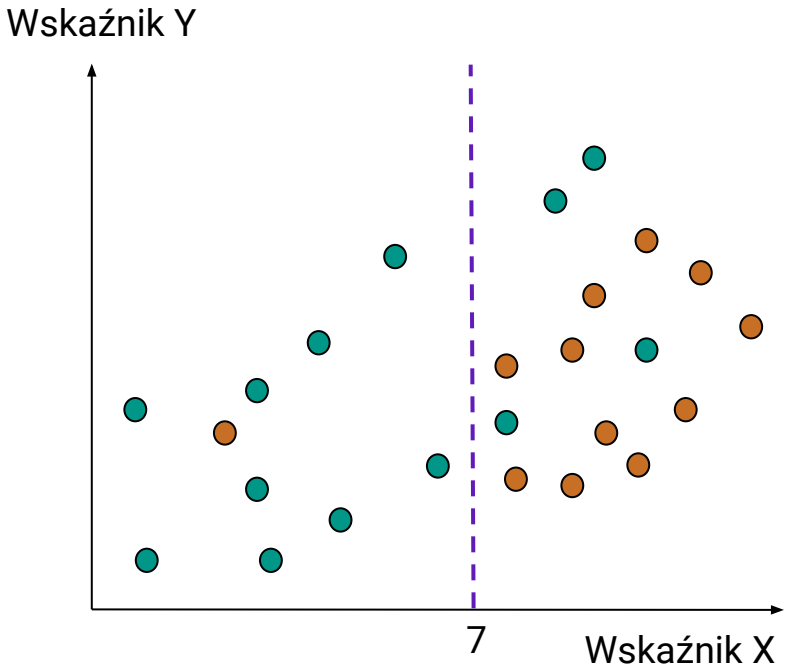
Drzewa decyzyjne

Punkt odcięcia	Gini dla kryterium
X >= a	0.48
X >= b	0.32
X >= c	**0.31**
X >= d	0.33

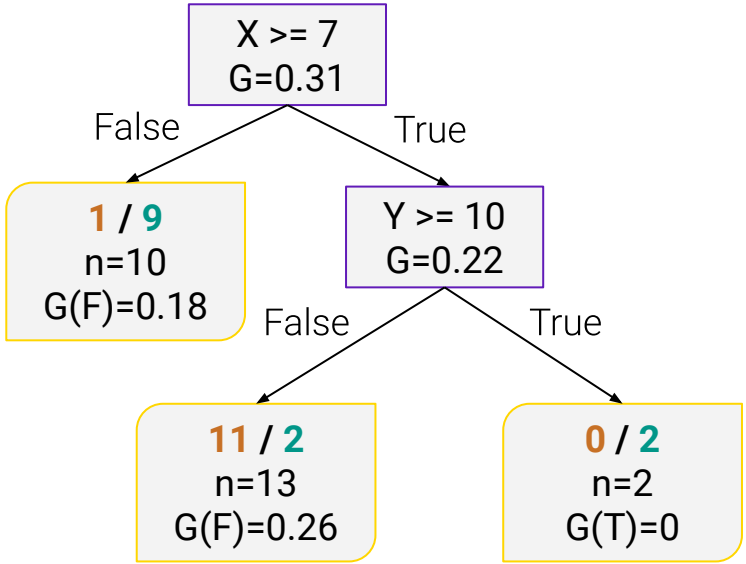


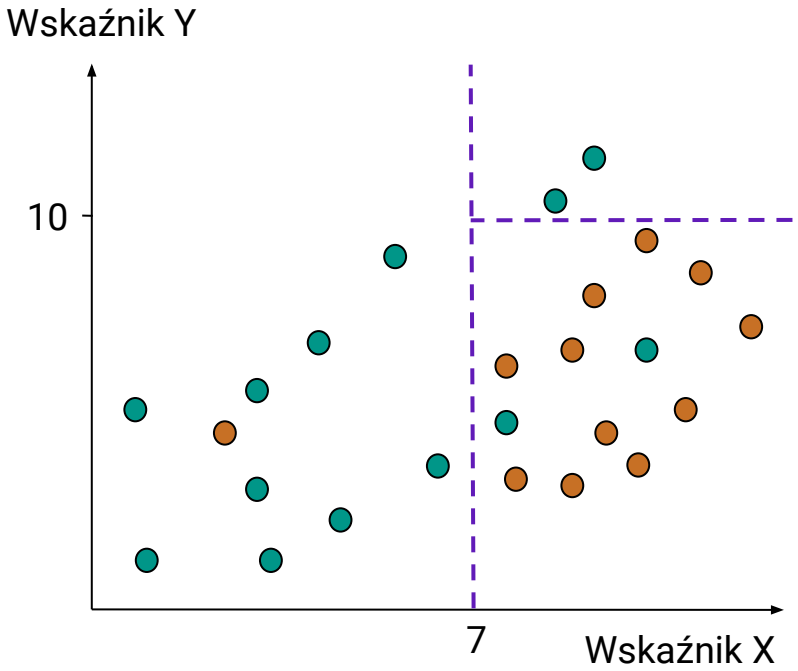
Drzewa decyzyjne



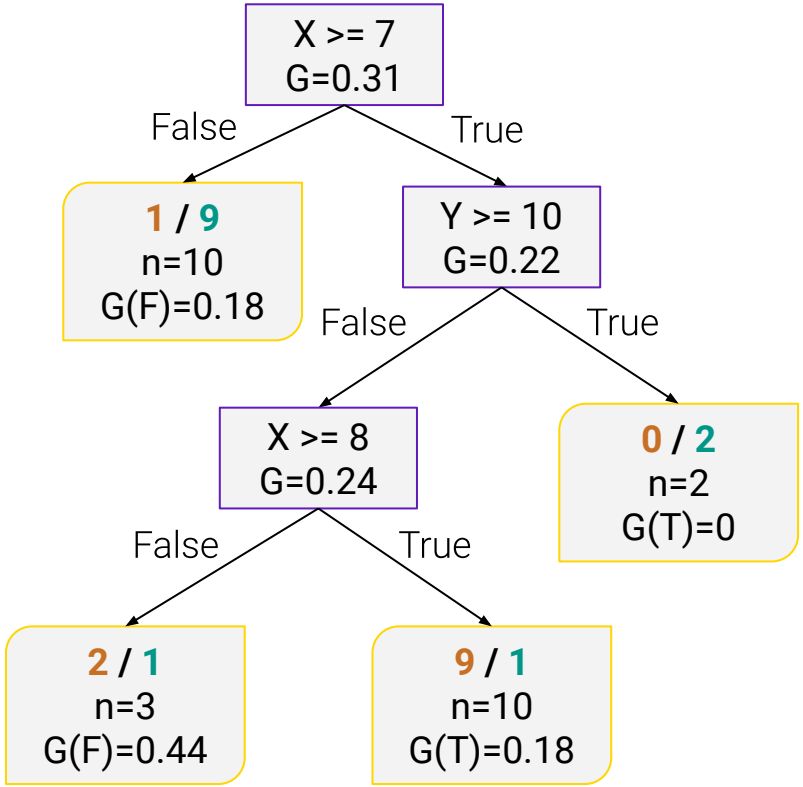


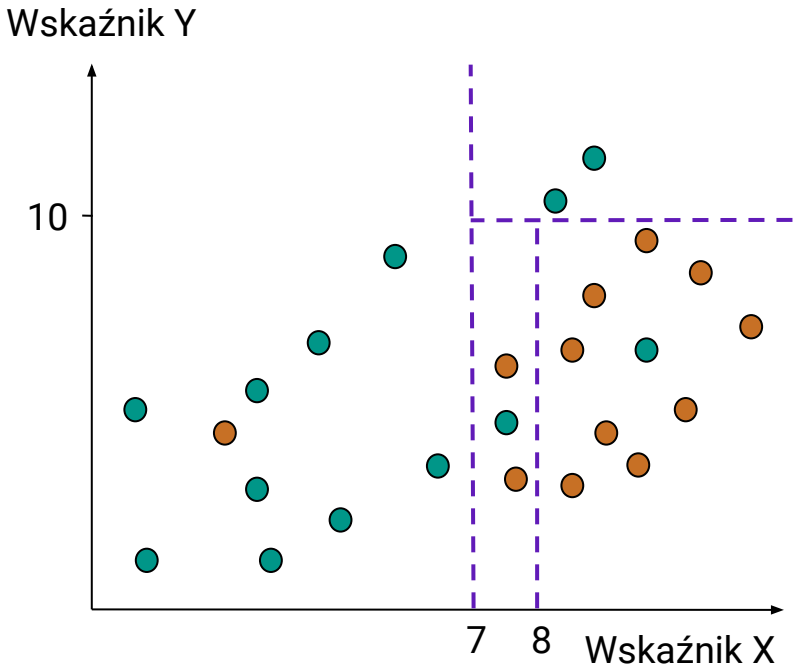
Drzewa decyzyjne



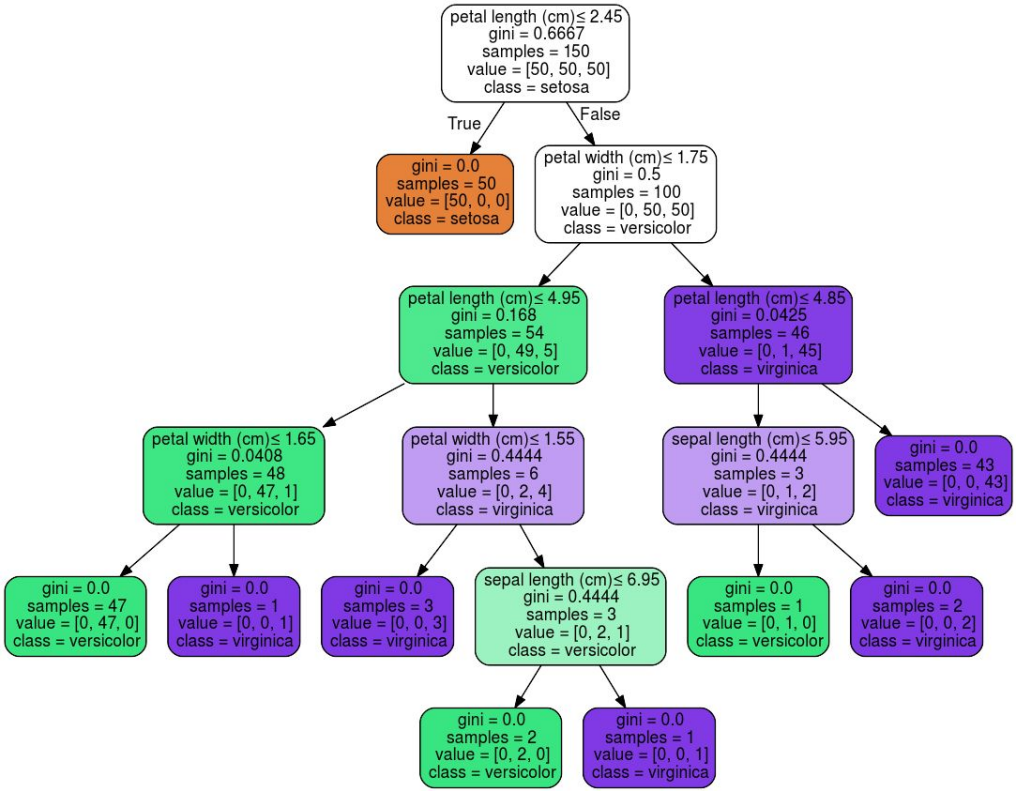


Drzewa decyzyjne





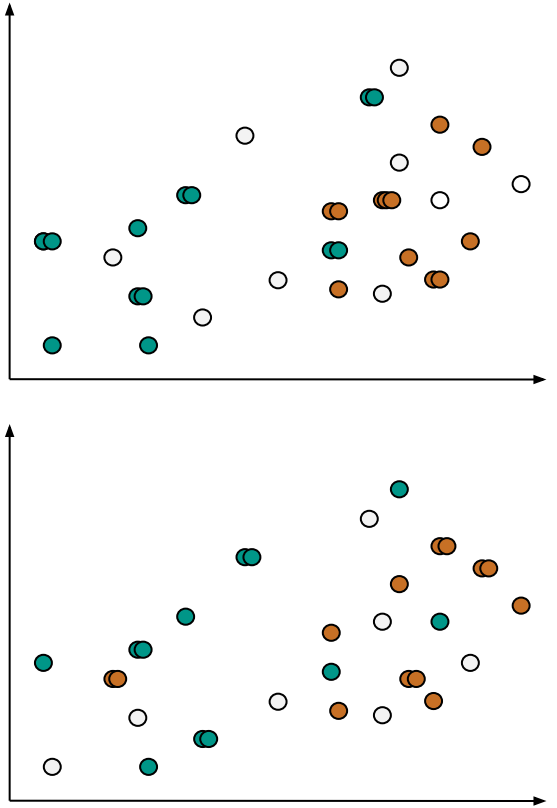
Drzewa decyzyjne



scikit-learn.org

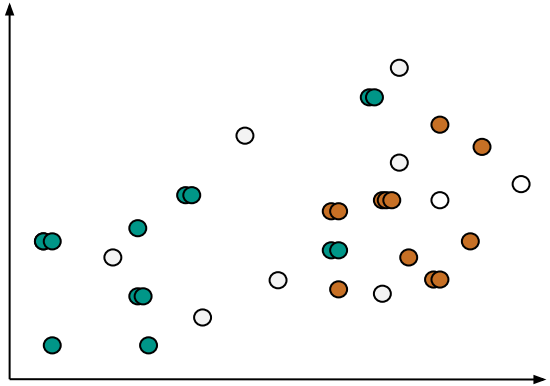
Wypróbujmy to!

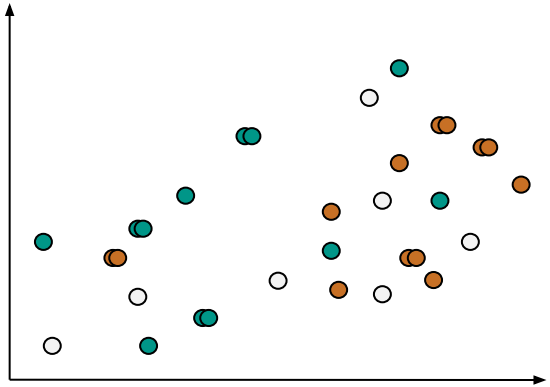
Lasy losowe (ang. Random Forests)

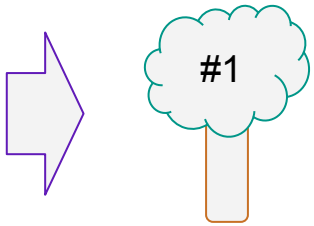


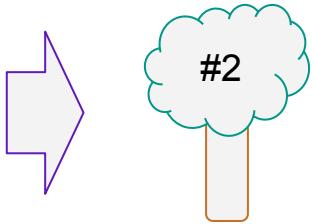
1. Dla każdego drzewa:
Z całego zbioru próbek o liczności **N** wylosuj **N** punktów z powtórzeniami (bootstraping).

Lasy losowe (ang. Random Forests)







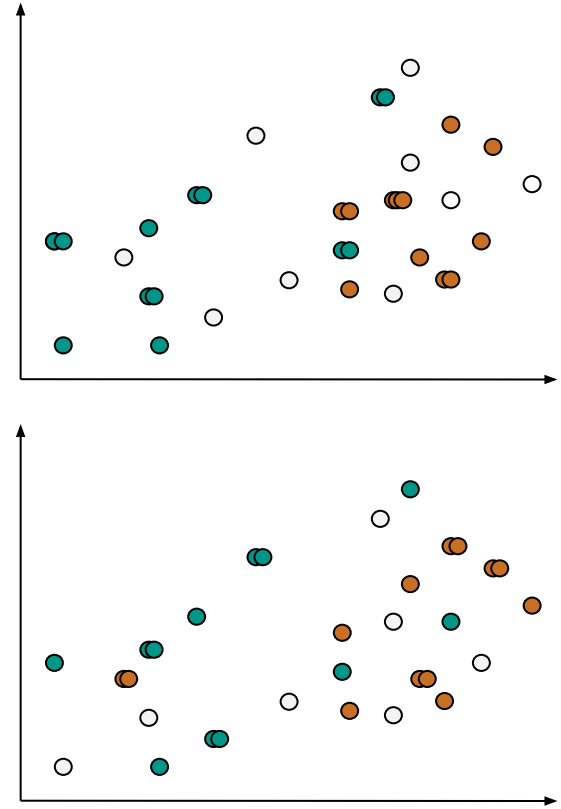


2. Dla każdego drzewa:
Wybierz losowo **K** cech, na podstawie których będzie budowane drzewo (parametr max_features)

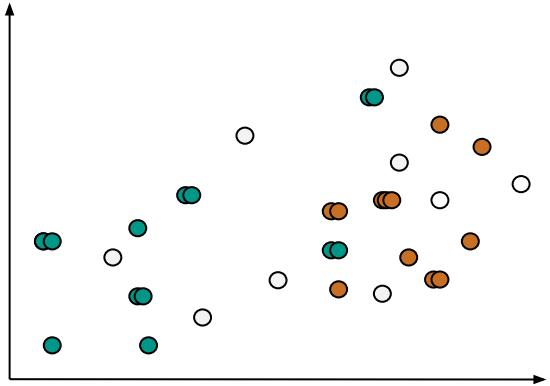
Dla K = 3 dla przykładowego drzewa:

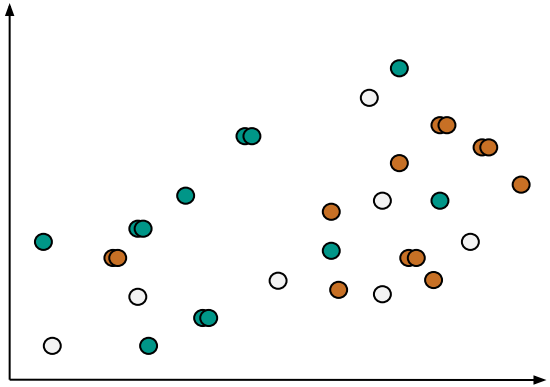
x1	x2	x3	x4	x5	y

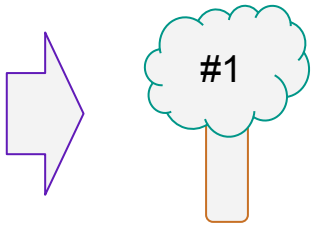
Lasy losowe (ang. Random Forests)

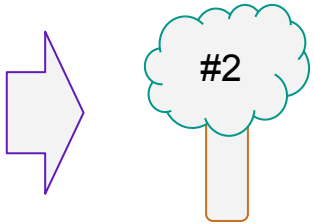


Lasy losowe (ang. Random Forests)









label #1
(True v False)

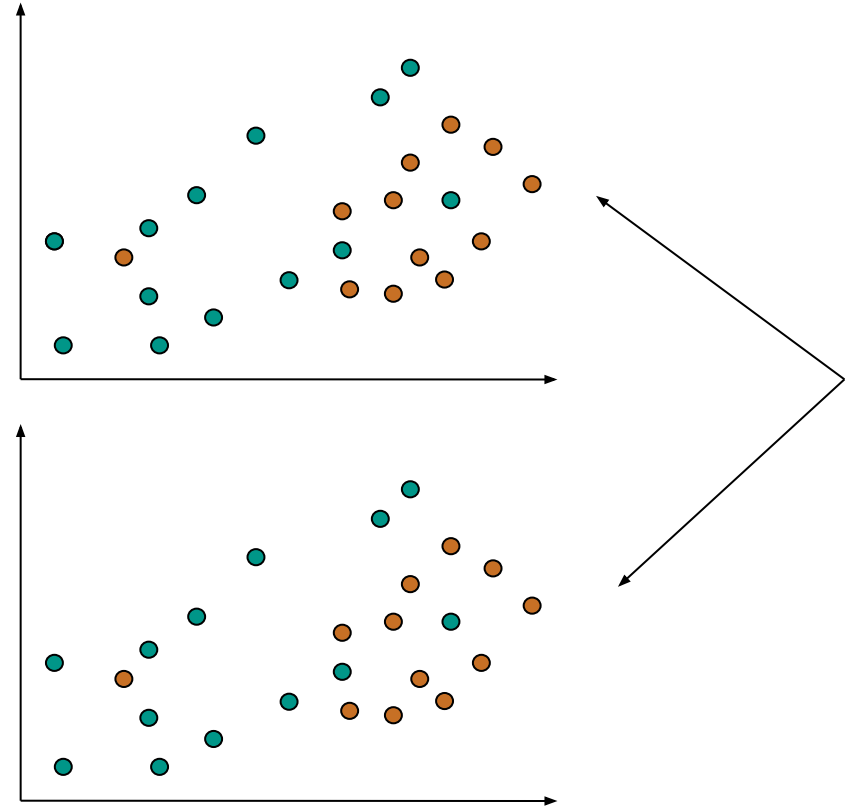
label #2
(True v False)

4. Głosowanie większościowe

finalny label
(True v False)

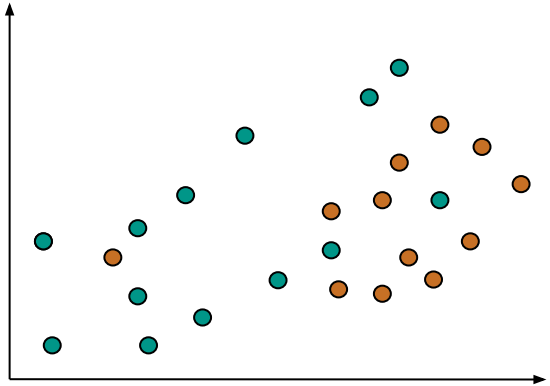
Wypróbujmy to!

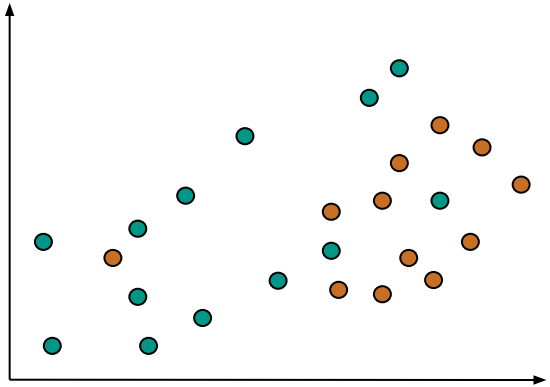
Ekstremalne lasy losowe (ang. Extra trees)

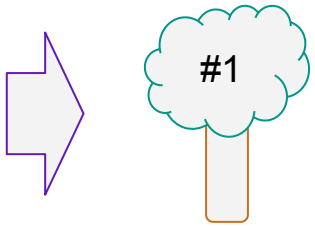


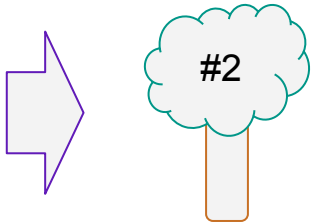
1. Dla każdego drzewa:
Wybierz cały dostępny zbiór jako dane treningowe.

Ekstremalne lasy losowe (ang. Extra trees)







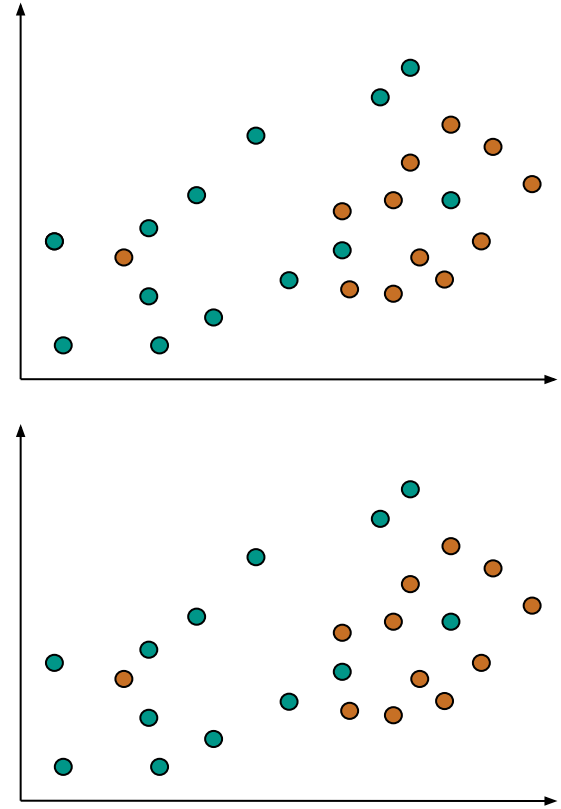


2. Dla każdego drzewa:
Dla każdego podziału w drzewie wybierz losowo **K** cech (parametr max_features).
3. Dla każdej z **K** cech wylosuj punkt odcięcia, a następnie wybierz ten, który daje najczystszy podział.

Dla K = 3 dla przykładowego drzewa:

x1	x2	x3	x4	x5	y

Ekstremalne lasy losowe (ang. Extra trees)



#1

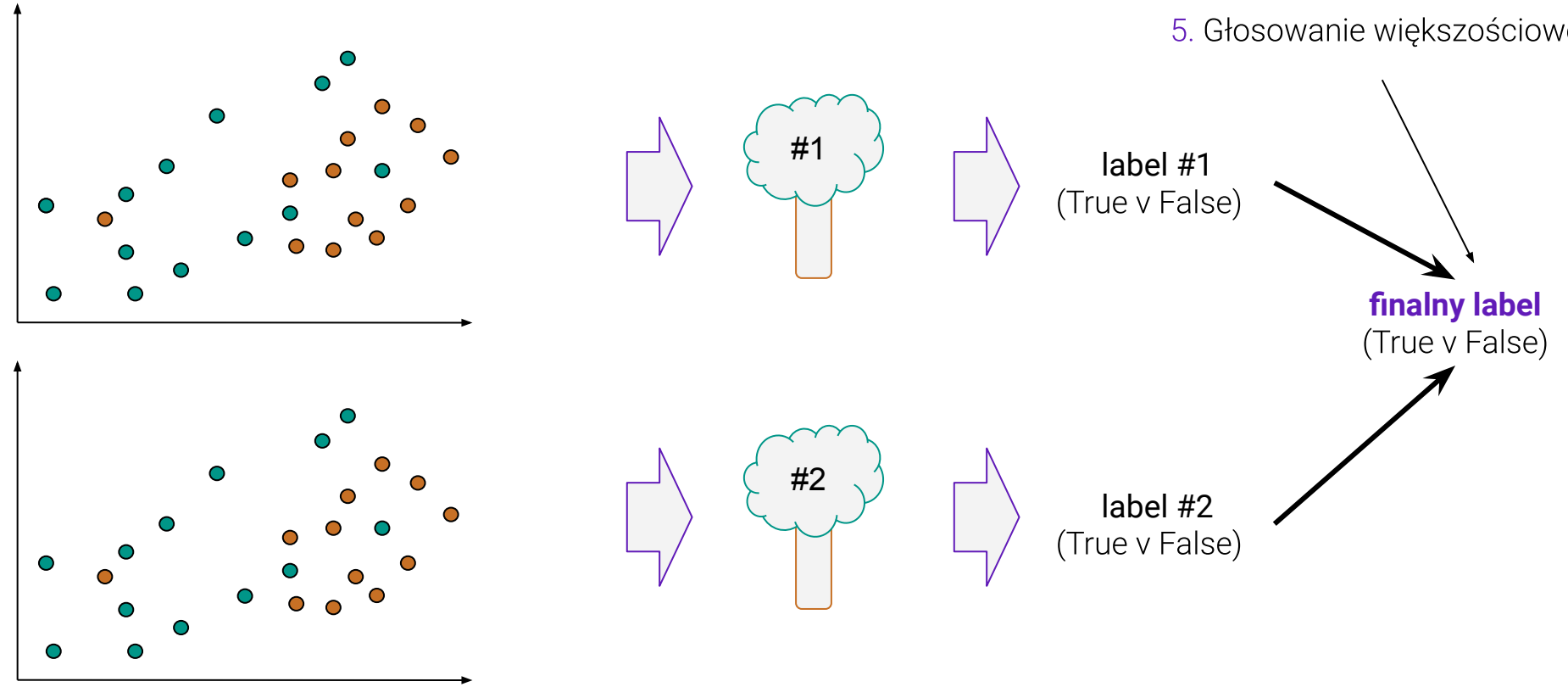
label #1
(True v False)

#2

label #2
(True v False)

4. Dla każdego drzewa:
Wykonaj predykcje.

Ekstremalne lasy losowe (ang. Extra trees)



Wypróbujmy to!

Miary oceny modeli przy klasyfikacji

Macierz pomyłek (ang. confusion matrix)

Predicted / Real	Pred: False	Pred: True
Real: False	**121** True Negative	**24** False Positive
Real: True	**7** False Negative	**87** True Positive

Przykłady dla klasyfikacji pacjentów (wykrywanie choroby):

True Negative (TN) - Ludzie **zdrowi** poprawnie zdiagnozowani jako **zdrowi**.

False Positive (FP) - Ludzie **zdrowi** błędnie zdiagnozowani jako **chorzy**.

False Negative (FN) - Ludzie **chorzy** błędnie zdiagnozowani jako **zdrowi**.

True Positive (TP) - Ludzie **chorzy** poprawnie zdiagnozowani jako **chorzy**.

Macierz pomyłek dla wielu klas

Real \ Predicted	HIV	Cancer	SM
HIV	**87**	**3**	**4**
Cancer	**5**	**64**	**7**
SM	**19**	**17**	**33**

Często zdarza się, że klas jest więcej niż jedna.

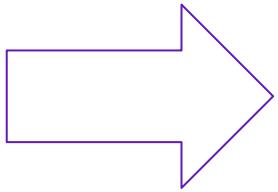
Pokazana macierz pomyłek reprezentuje sytuację, w której chory pacjent musi zostać sklasyfikowany pod względem choroby, którą posiada.

Możliwe choroby to:
- ➔ HIV,
- ➔ rak,
- ➔ stwardnienie rozsiane.

Zakładamy, że pacjent jest chory na jedną i tylko jedną chorobę.

Macierz pomyłek dla wielu klas

Real \ Predicted	HIV	Cancer	SM
HIV	**87**	**3**	**4**
Cancer	**5**	**64**	**7**
SM	**19**	**17**	**33**



Binarna macierz pomyłek dla HIV:

Real \ Predicted	Pred: False	Pred: True
Real: False	**121** True Negative	**24** False Positive
Real: True	**7** False Negative	**87** True Positive

Błędy klasyfikacji

Dokładność (ang. accuracy, ACC)

Prawdopodobieństwo dokonania poprawnej klasyfikacji dla losowo wybranej binarnej próbki.
ACC = (TP + TN) / (FP + TP + FN + TN)

Przykład (dla klasyfikacji HIV/nie HIV):

Prawdopodobieństwo, że losowo wybrany pacjent spośród wszystkich został poprawnie zaklasyfikowany.
ACC = (87 + 121) / (24 +87 + 7 + 121) = 0.871

Predicted / **Real**	Pred: False	Pred: True
Real: False	**121** True Negative	**24** False Positive
Real: True	**7** False Negative	**87** True Positive

Błędy klasyfikacji

Precyzja (ang. precision, positive predictive value, PPV)

Prawdopodobieństwo, iż losowa binarna próbka zaklasyfikowana jako True, jest rzeczywiście True.
PPV = TP / (TP + FP)

Przykład (dla klasyfikacji HIV/nie HIV):

Prawdopodobieństwo, że losowo wybrany pacjent spośród zaklasyfikowanych jako chory na HIV jest rzeczywiście chory na HIV.
PPV = 87 / (87 + 24) = 0.783

Predicted / **Real**	Pred: False	Pred: True
Real: False	**121** True Negative	**24** False Positive
Real: True	**7** False Negative	**87** True Positive

Błędy klasyfikacji

Czułość (ang. sensitivity, recall, true positive rate, TPR)

Prawdopodobieństwo, iż losowa binarna próbka rzeczywista True, została zaklasyfikowana jako True.
TPR = TP / (TP + FN)

Przykład (dla klasyfikacji HIV/nie HIV):

Prawdopodobieństwo, że losowo wybrany pacjent spośród chorych na HIV został poprawnie zaklasyfikowany jako chory na HIV.
TPR = 87 / (87 + 7) = 0.925

Predicted / Real	Pred: False	Pred: True
Real: False	**121** True Negative	**24** False Positive
Real: True	**7** False Negative	**87** True Positive

Błędy klasyfikacji

Specyficzność (ang. fall-out, false positive rate, FPR)

Prawdopodobieństwo, iż losowa binarna próbka rzeczywista False, została źle zaklasyfikowana jako True.
FPR = FP / (FP + TN)

Przykład (dla klasyfikacji HIV/nie HIV):

Prawdopodobieństwo, że losowo wybrany pacjent spośród niechorych na HIV został niepoprawnie zaklasyfikowany jako chory na HIV.
FPR = 24 / (24 + 121) = 0.166

Predicted / Real	Pred: False	Pred: True
Real: False	**121** True Negative	**24** False Positive
Real: True	**7** False Negative	**87** True Positive

Błędy dla wielu klas, mikro- i makrouśrednianie

Na przykładzie miary ACC

	TP	TN	FP	FN	ACC
dla klasy **HIV**	87	121	24	7	0.871
dla klasy **Cancer**	64	143	20	12	0.866
dla klasy **SM**	33	159	11	36	0.842
wartość średnia	61.3	141	18.3	18.3	0.86

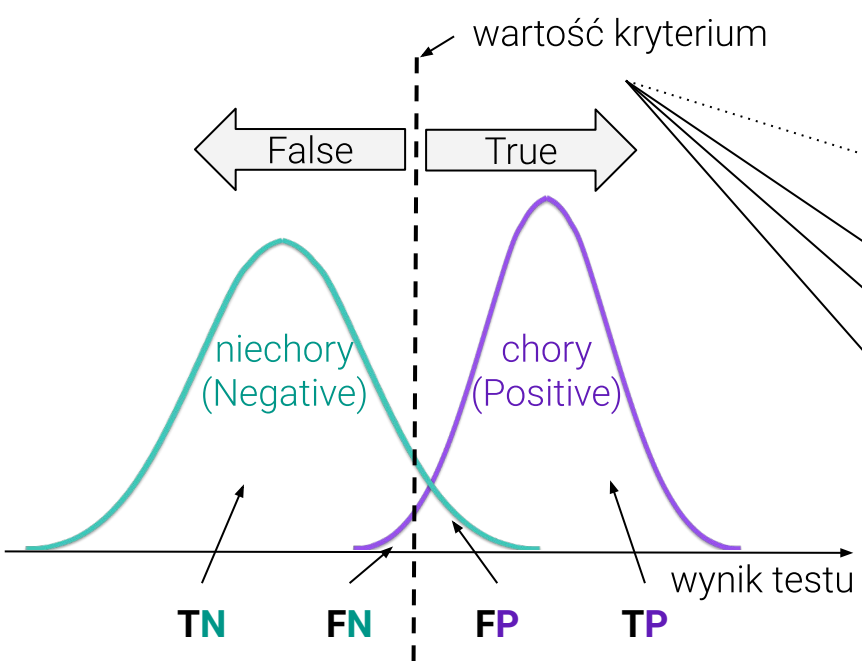
ACC_{makro}

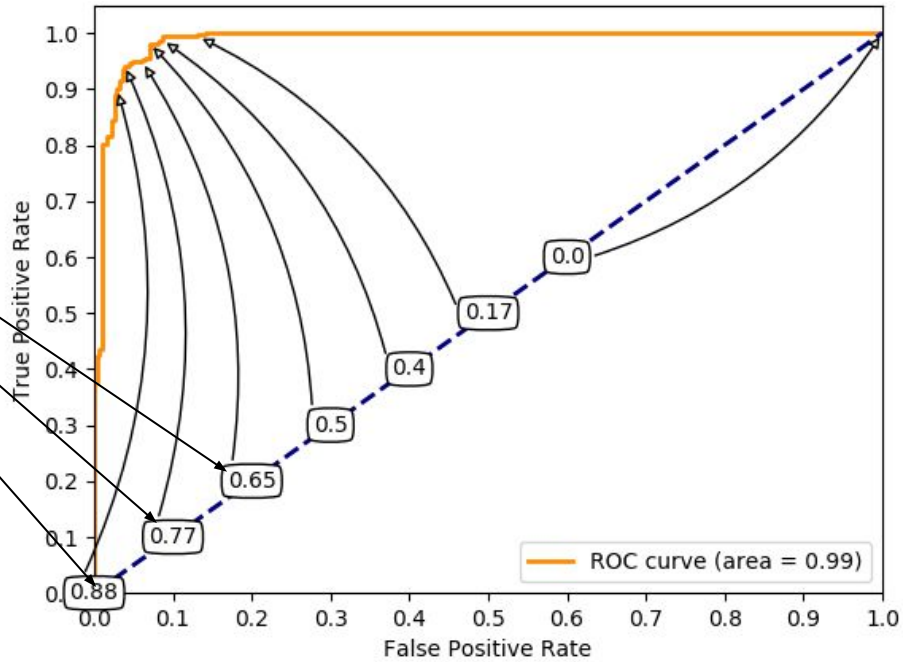
Mikrouśrednianie:
- jest wrażliwe na nierówne liczności klas
- mierzy "wkład" każdej z klas

Makrouśrednianie:
- traktuje wszystkie klasy jednakowo
- jest zwykłą średnią arytmetyczną z miary dokładności

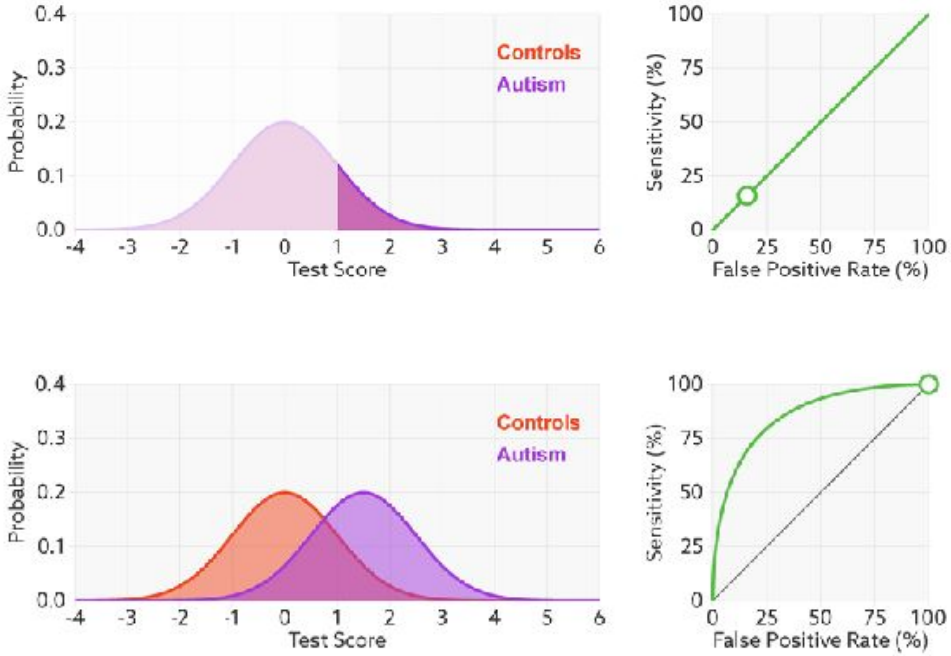
$ACC_{makro} = (61.3 + 141) / (61.3 + 141 + 18.3 + 18.3) = 0.847$

Krzywa ROC


wartość kryterium
False
True
niechory
(Negative)
chory
(Positive)
wynik testu
TN
FN
FP
TP


True Positive Rate
False Positive Rate
0.0
0.17
0.4
0.5
0.65
0.77
0.88
ROC curve (area = 0.99)

Krzywa ROC

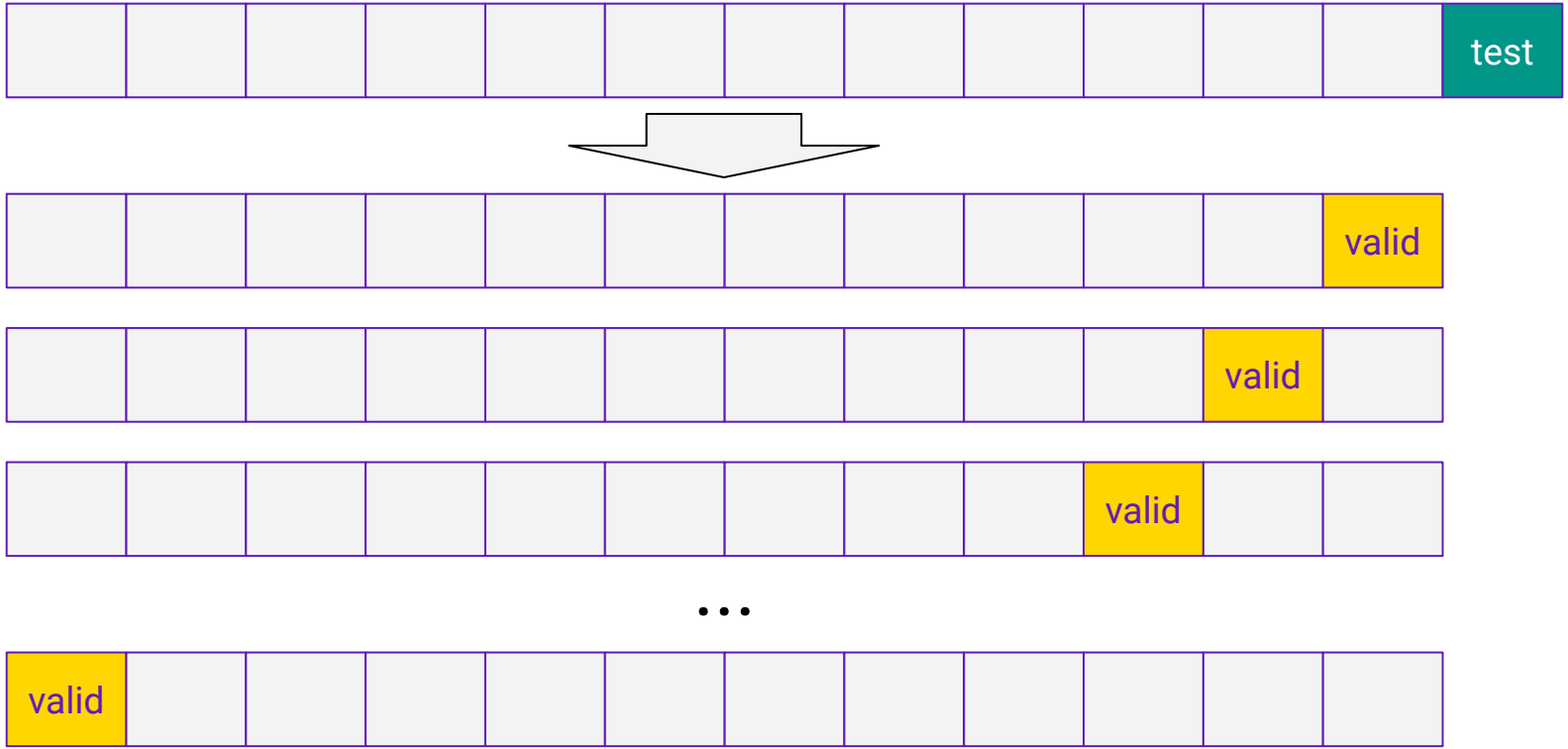


lukeoakdenrayner.wordpress.com

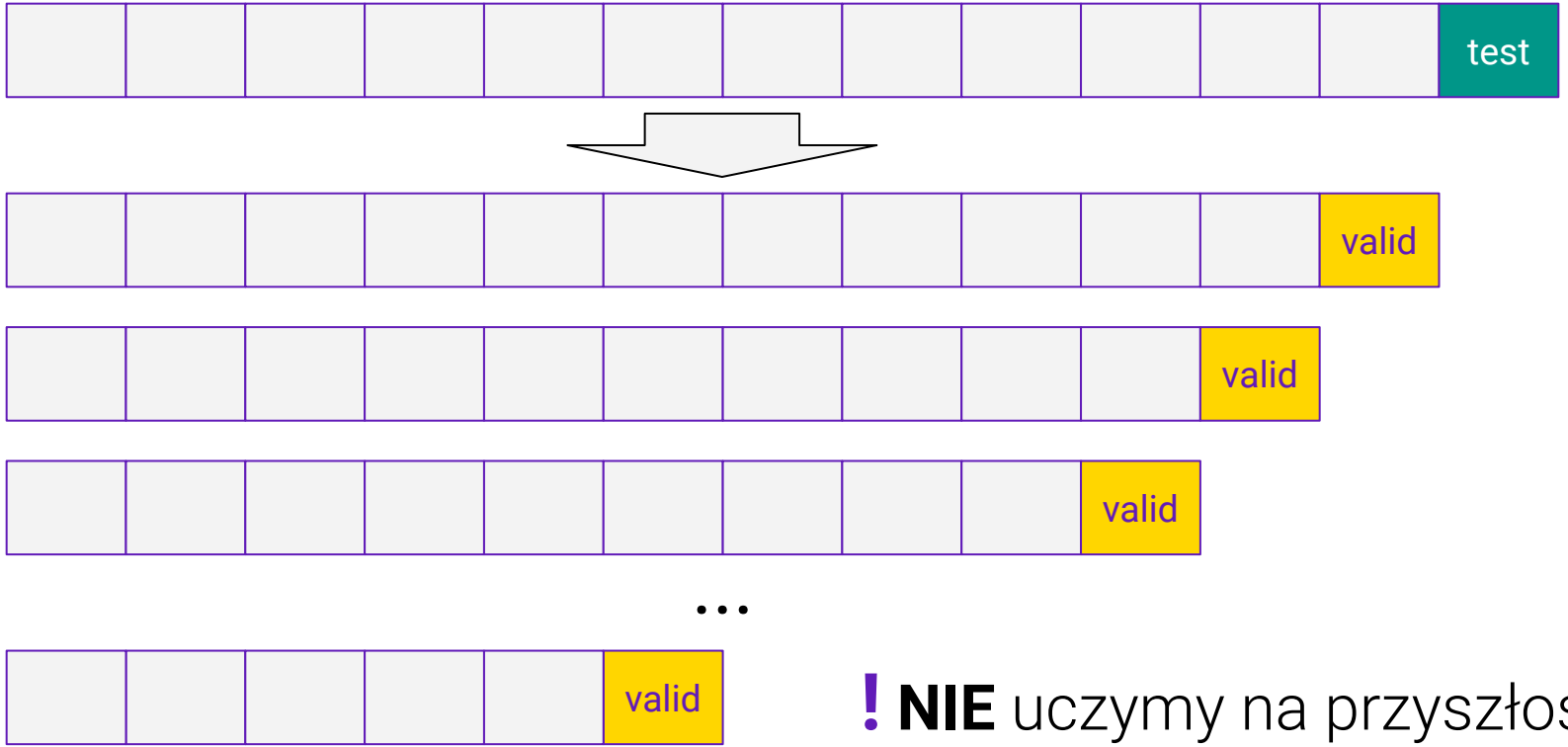
Wypróbujmy to!

Dobór modeli i dobre praktyki

Walidacja krzyżowa (ang. cross validation)

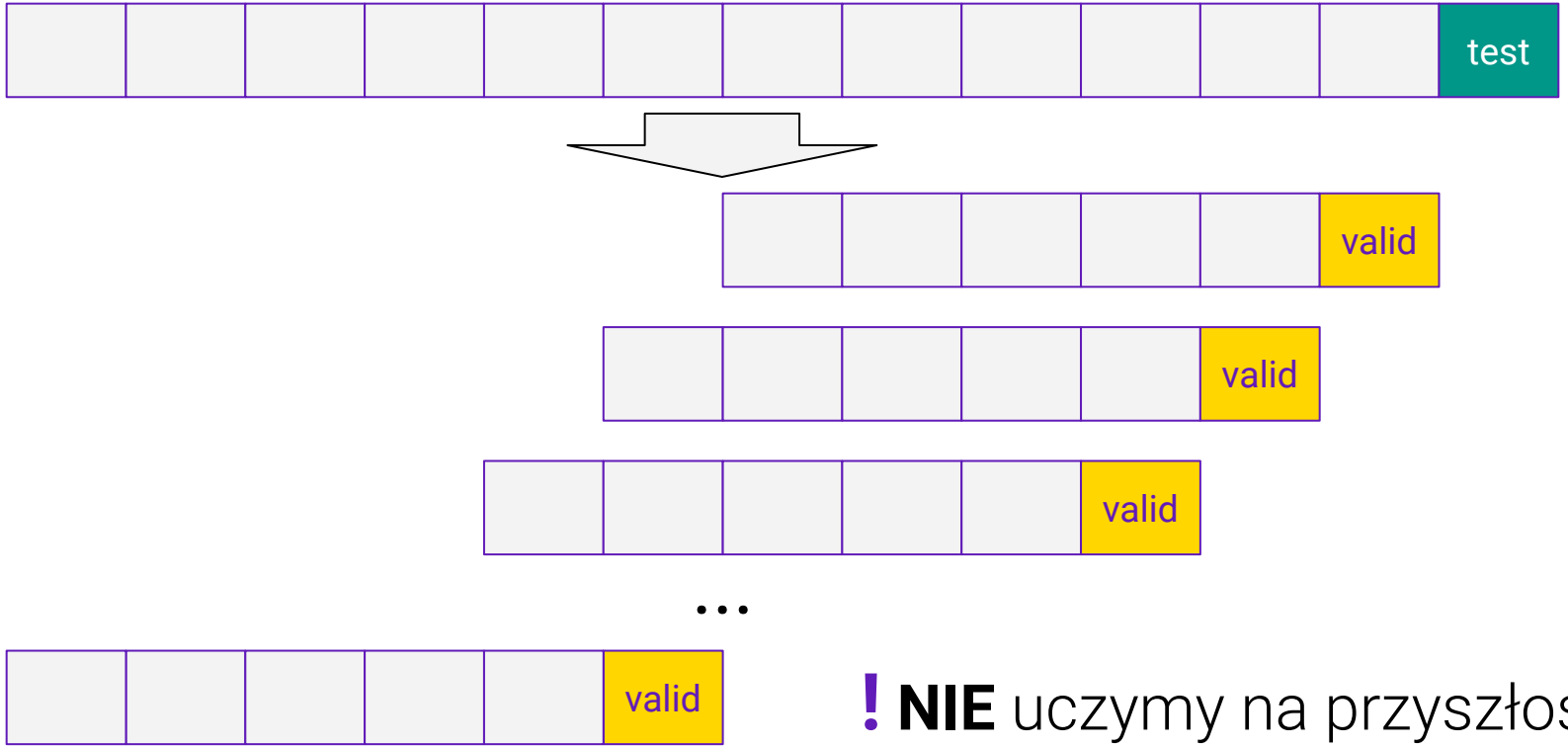


Walidacja krzyżowa dla szeregów czasowych 1



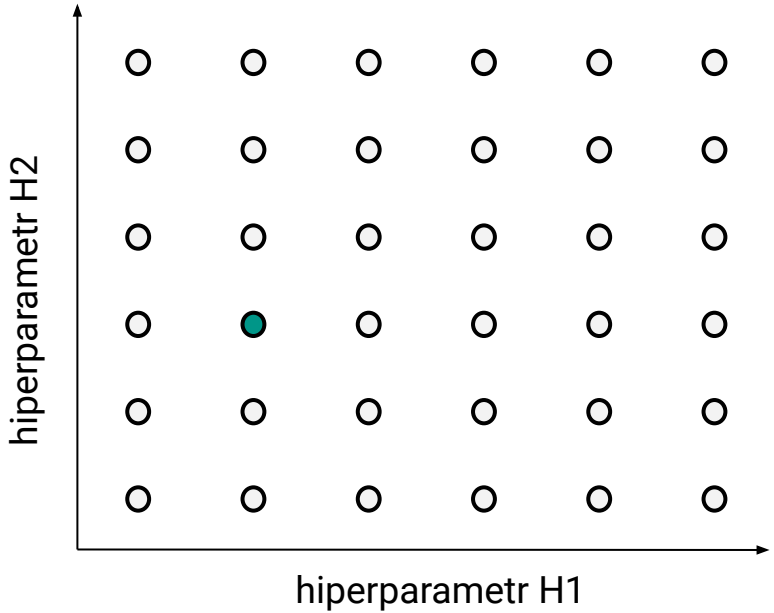
! **NIE** uczymy na przyszłości

Walidacja krzyżowa dla szeregów czasowych 2



! **NIE** uczymy na przyszłości

Dobór modelu - Grid Search



1. Wygeneruj w wielowymiarowej przestrzeni **wszystkie** kombinacje z **list** wartości hiperparametrów, tworząc tym samym **siatkę** (grid) wszystkich (**n**) możliwości.
2. Stwórz **n** modeli i wybierz **najlepszy** punkt (z **najmniejszym** błędem modelu).

Dobór modelu - Grid Search

Zalety:

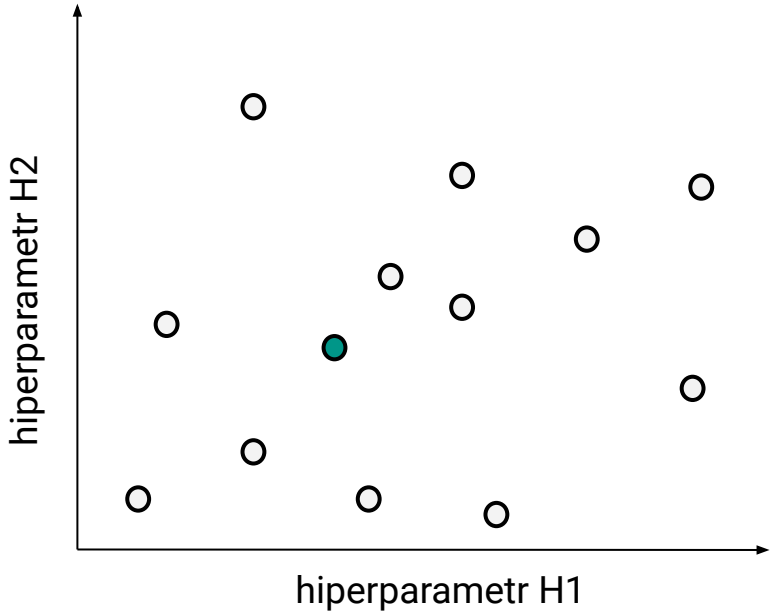
- z dużym prawdopodobieństwem wskazuje optymalny model,
- nie wymaga iteracji,
- wygodny przy dyskretnym skończonym zbiorze wartości danego hiperparametru.

Wady:

- dla wielowymiarowej (ciągłej) przestrzeni hiperparametrów ogromna złożoność obliczeniowa,
- wymaga podanie explicite listy wartości hiperparametrów, które przeszukujemy.

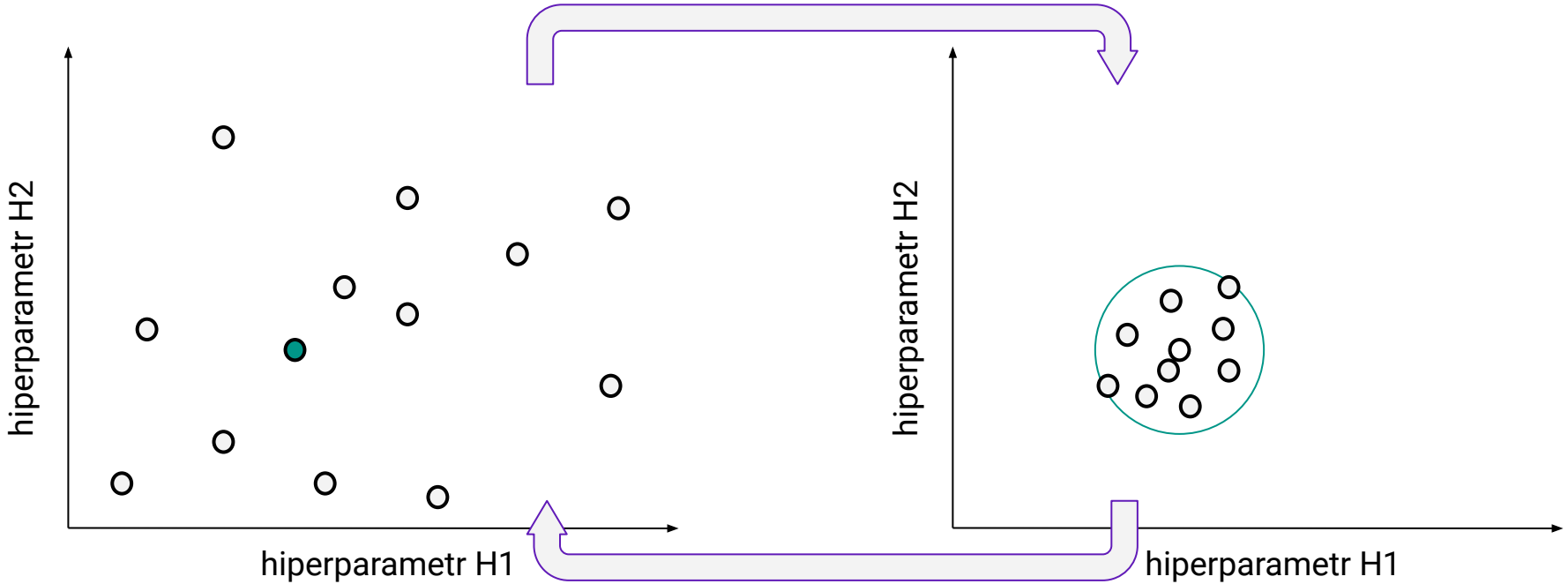
Dobór modelu - Random Search + Hill Climbing

1. Wygeneruj losowo w zadanej przestrzeni hiperparametrów **n** punktów.
2. Wybierz **najlepszy** punkt (z **najmniejszym** błędem modelu).



Dobór modelu - Random Search + Hill Climbing

3. Wygeneruj losowo **n** punktów w bliskiej odległości od **najlepszego** punktu.
4. Wróć do **punktu 2**, powtórz **kilkukrotnie** procedurę aż do uzyskania pożądanego poziomu błędu modelu.



Dobór modelu - Random Search

Zalety:

- dla wielowymiarowej (ciągłej) przestrzeni hiperparametrów mała złożoność obliczeniowa,
- nie wymaga podania listy wartości hiperparametrów, które przeszukujemy, wystarczą zakresy,
- w większości bardziej złożonych modeli sprawdza się lepiej niż Grid Search.

Wady:

- mniejsza szansa (w porównaniu z Grid Search) na znalezienie optymalnego modelu,
- wymaga iteracji,
- utrudnione losowanie dla hiperparametrów o dyskretnych wartościach.

Wypróbujmy to!

Podsumowanie i zakończenie

Co dalej?

Gdzie bywać, kogo znać?

Książki, które polecam na początek:

1. *“An Introduction to Statistical Learning with Applications in R”*, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani - dostępna online
2. *“The Elements of Statistical Learning Data Mining, Inference, and Prediction”*, Trevor Hastie, Robert Tibshirani, Jerome Friedman - dostępna online
3. *“Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”*, Aurélien Géron
4. *“Deep Learning”* Ian Goodfellow, Yoshua Bengio, Aaron Courville
5. *“Reinforcement Learning: An Introduction”*, Richard S. Sutton, Andrew G. Barto - dostępna online

Kursy, które polecam na początek:

1. *“Machine Learning”*, Coursera, Stanford University, Andrew Ng, sprawdź
2. *“Deep Learning Specialization”*, Coursera, deeplearning.ai, Andrew Ng, sprawdź
3. Cokolwiek, Andrew Ng
4. *“Artificial Intelligence”*, YouTube, MIT OpenCourseWare, sprawdź
5. “*RL Course by David Silver*”, YouTube, sprawdź

Blogi, ludzie i wydarzenia, które warto śledzić:

1. wildml.com
2. machinelearningmastery.com
3. towardsdatascience.com
4. medium.com
5. colah.github.io
6. Andrew Ng
7. Geoffrey Hinton
8. Confitura
9. PyData
10. Data Science Summit

Ankieta

tinyurl.com/introduction-to-ml-confitura