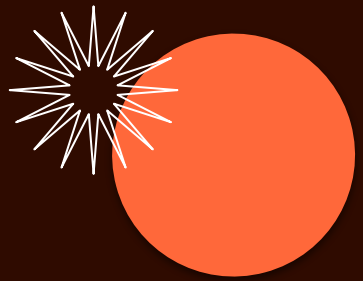
POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

METODY ANALIZY I EKSPLORACJI DANYCH

Wykład 8 - Predykcja: Klasyfikacja

DR INŻ. AGATA MIGALSKA



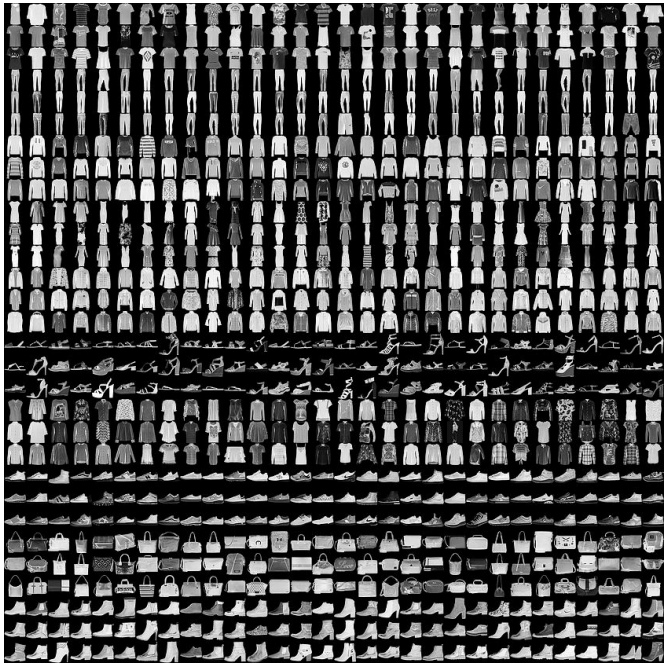
00

CEL I MOTYWACJA

KLASYFIKACJA

- Obszary zastosowań:
  - Klasyfikacja wg gatunków
  - Identyfikacja biometryczna
  - Wizja komputerowa
  - Analiza obrazu medycznego i obrazowanie medyczne
  - Rozpoznawanie pisma odręcznego
  - Klasyfikacja dokumentów
  - Prognoza odejścia klienta
  - Ocena zdolności kredytowej
  - Wykrywanie oszustw bankowych

PROBLEM KLASYFIKACJI

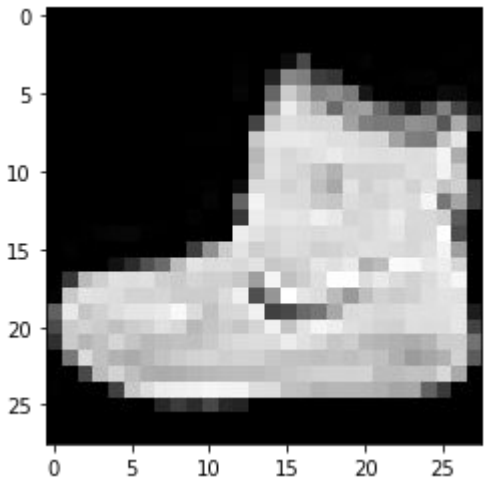


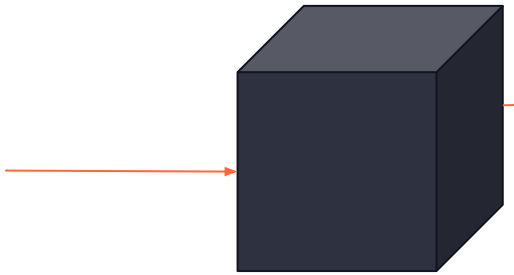
Klasyfikacja ubrań wg typu

- Przykładowy zbiór: Fashion-MNIST
- obrazy w odcieniach szarości
- o rozdzielczości 28x28
- Zbiór powiązany z etykietami z 10 klas:
  - 0 - t-shirt/top
  - 1 - spodnie
  - 2 - sweter
  - 3 - sukienka
  - 4 - płaszcz
  - 5 - sandał
  - 6 - koszula
  - 7 - trampki
  - 8 - torba
  - 9 - but do kostki

Źródło: https://github.com/zalandoresearch/fashion-mnist

PROBLEM KLASYFIKACJI





9 - but do kostki

NIEZNANA OBSERWACJA

MODEL

ETYKIETA

01 K-NAJBLIŻSZYCH SĄSIADÓW

02 DOBROĆ KLASYFIKATORA

03 REGRESJA LOGISTYCZNA

04 MASZYNY WEKTORÓW NOŚNYCH

05 DRZEWA DECYZYJNE

06 SIECI NEURONOWE

01

K-NAJBLIŻSZYCH SĄSIADÓW

KLASYFIKACJA OWOCÓW

KLASA

CECHY

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79

OBSERWACJA / PRÓBKA / KROTKA

PODZIAŁ ZBIORU DANYCH

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79
...	...	...	...	...	...	...	...
54	4	lemon	unknown	116	6.1	8.5	0.71
55	4	lemon	unknown	116	6.3	7.7	0.72
56	4	lemon	unknown	116	5.9	8.1	0.73
57	4	lemon	unknown	152	6.5	8.5	0.72
58	4	lemon	unknown	118	6.1	8.1	0.70

KLASYFIKATOR K-NAJBLIŻSZYCH SĄSIADÓW

Dane:

- Zbiór uczący X_{train} i zbiór odpowiadających mu etykiet y_{train}.
- Nowa obserwacja x_{test}, która ma być zaklasyfikowana.

Algorytm:

1. Znajdź w X_{train} k obserwacji, które są najbardziej podobne do x_{test} - nazwijmy ten zbiór X_{NN}.
2. Pobierz etykiety y_{NN} odpowiadające zbiorowi X_{NN}.
3. Wyznacz etykietę dla obserwacji x_{test} na podstawie etykiet y_{NN} np. poprzez głosowanie większościowe.

KLASYFIKATOR K-NAJBLIŻSZYCH SĄSIADÓW

Dane:

- Zbiór uczący X_{train} i zbiór odpowiadających mu etykiet y_{train}.
- Nowa obserwacja x_{test}, która ma być zaklasyfikowana.

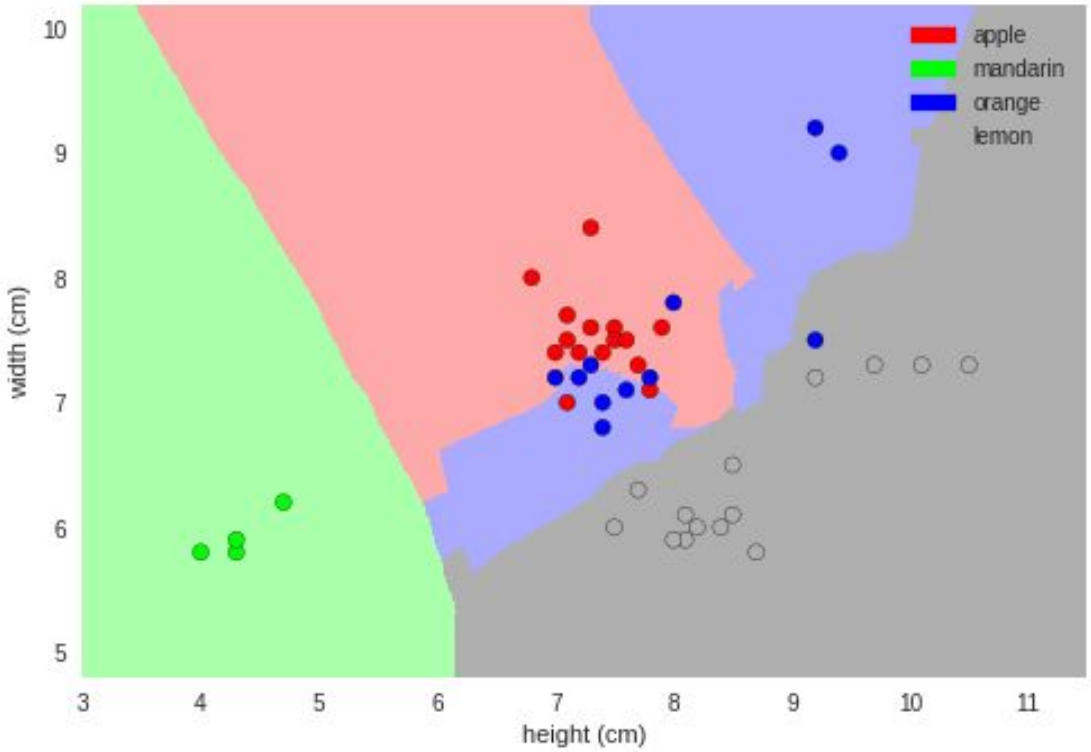
Algorytm:

1. Znajdź w X_{train} k obserwacji, które są najbardziej podobne do x_{test} - nazwijmy ten zbiór X_{NN}.
2. Pobierz etykiety y_{NN} odpowiadające zbiorowi X_{NN}.
3. Wyznacz etykietę dla obserwacji x_{test} na podstawie etykiet y_{NN} np. poprzez głosowanie większościowe.

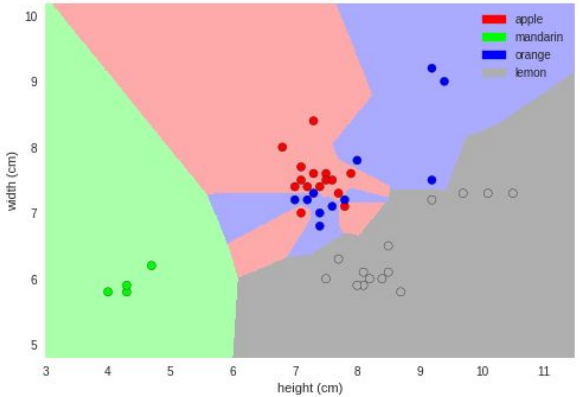
1. Metryka
2. Liczba "najbliższych" sąsiadów do analizy
3. Metoda agregująca etykiety sąsiadów
4. Opcjonalnie funkcja ważąca sąsiadów (np. im dalszy sąsiad tym mniejszy jego wpływ na wynik)

GRANICE DECYZYJNE

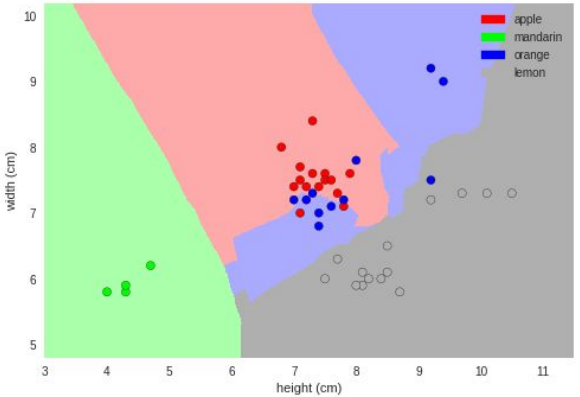
K=5



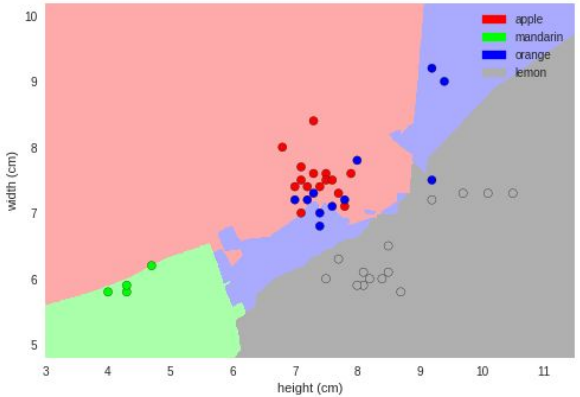
GRANICE DECYZYJNE



K=1



K=5



K=10

TABLICA POMYŁEK (MACIERZ BŁĘDÓW)

		Klasa predykowana	
		Klasyfikacja pozytywna	Klasyfikacja negatywna
Klasa rzeczywista	Stan pozytywny	prawdziwie dodatnia, TP	prawdziwie ujemna, FN (błąd II rodzaju)
	Stan negatywny	fałszywie dodatnia, FP (błąd I rodzaju)	prawdziwie ujemna, TN

DOKŁADNOŚĆ KLASYFIKACJI OWOCÓW

	apple	mandarin	orange	lemon
apple	3	0	0	1
mandarin	0	1	0	0
orange	3	0	3	2
lemon	0	0	1	1

Klas

Prawda \	apple	~apple
apple	3	1
~apple	3	8

$$\text{Dokładność} = \frac{\text{poprawnie sklasyfikowane}}{\text{wszystkie obserwacje}}$$ TP+TN

$\text{wszystkie obserwacje} = 15$

$\text{poprawnie sklasyfikowane} = 8$

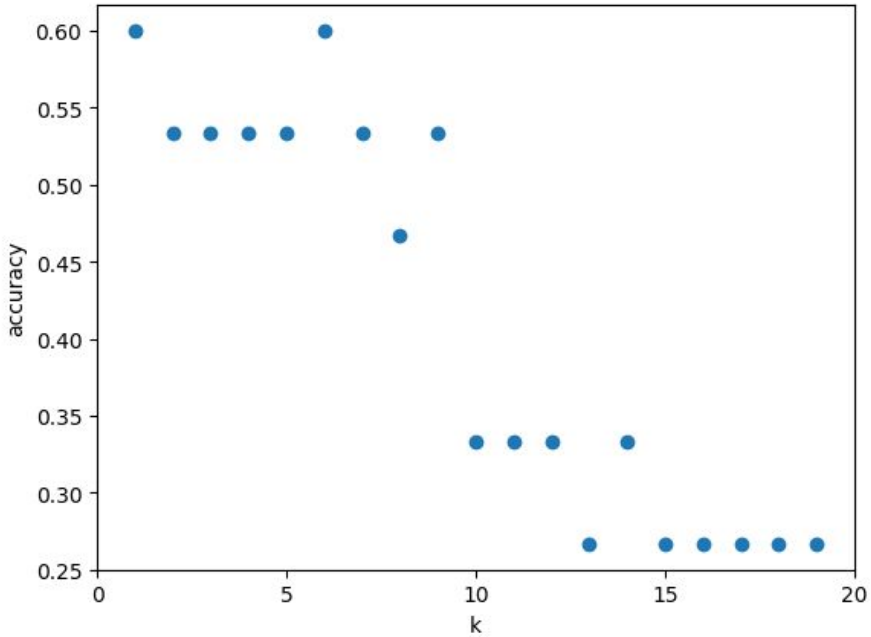
$\text{dokładność} = \frac{8}{15} = 0.5(3)$

wszystkie 15

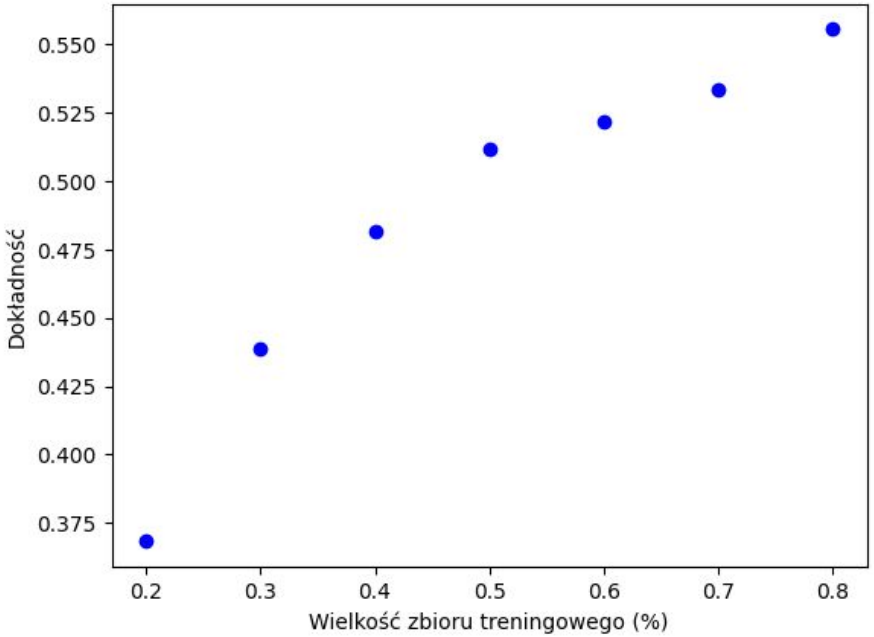
TP+TN = 11

Acc = 11/15

LICZBA SĄSIADÓW A DOKŁADNOŚĆ



WIELKOŚĆ ZBIORU UCZĄCEGO A DOKŁADNOŚĆ



Zbyt mało danych treningowych może spowodować, że model będzie zbyt dopasowanych do danych i nie będzie dobrze generalizować.

02

JAKOŚĆ / DOBROĆ KLASYFIKATORA

TABLICA POMYŁEK (MACIERZ BŁĘDÓW)

		Klasa predykowana	
		Klasyfikacja pozytywna	Klasyfikacja negatywna
Klasa rzeczywista	Stan pozytywny	prawdziwie dodatnia, TP	prawdziwie ujemna, FN (błąd II rodzaju)
	Stan negatywny	fałszywie dodatnia, FP (błąd I rodzaju)	prawdziwie ujemna, TN

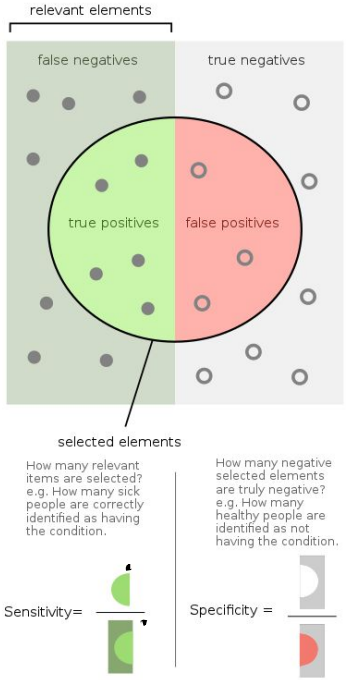
$$\text{Dokładność} = \frac{TP + TN}{TP + FP + TN + FN}$$

CZUŁOŚĆ I SWOISTOŚĆ

- Czułość (ang. sensitivity), TPR - prawdopodobieństwo, że stan pozytywny zostanie zaklasyfikowany jako pozytywny
- Swoistość (ang. specificity), FPR - prawdopodobieństwo, że stan negatywny zostanie zaklasyfikowany jako negatywny

$$\text{Czułość} = \frac{TP}{TP + FN}$$

$$\text{Swoistość} = \frac{TN}{TN + FP}$$

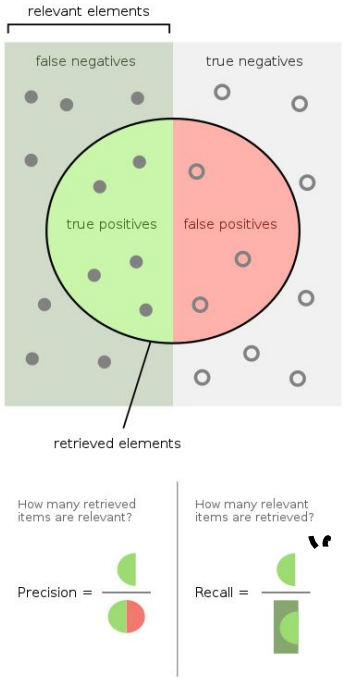


PRECYZJA I ZWROT (RECALL)

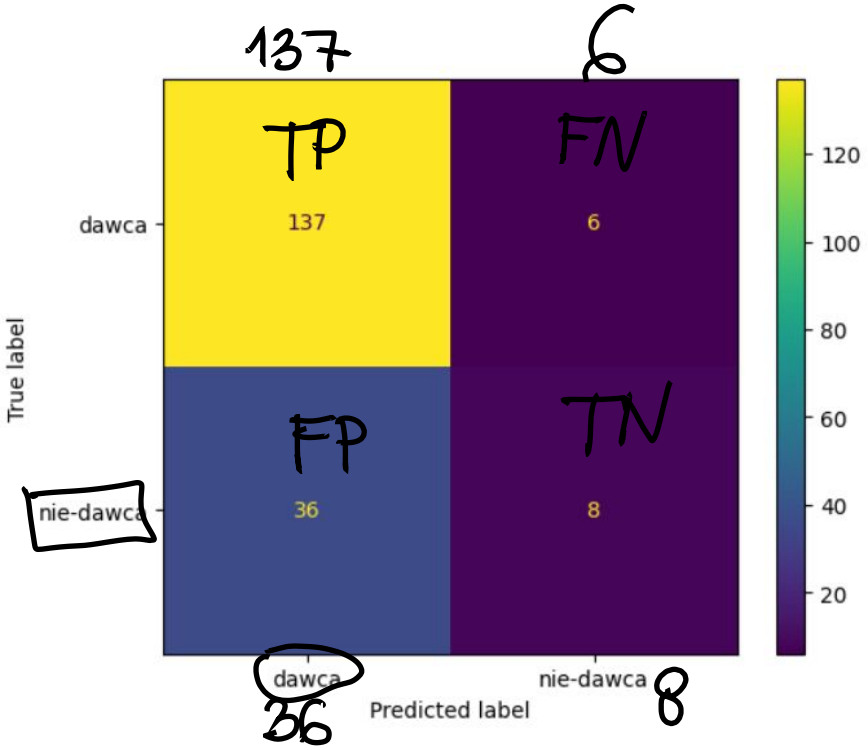
- Precyzja (ang. precision) - odsetek pozytywnych stanów zaklasyfikowanych jako pozytywne wśród wszystkich pozytywnych stanów
- Zwrot (ang. recall) - inaczej czułość - odsetek pozytywnych stanów zaklasyfikowanych jako pozytywne wśród pozytywnych klasyfikacji

$$\text{Precyzja} = \frac{TP}{TP + FP}$$

$$\text{Zwrot} = \frac{TP}{TP + FN}$$



PRZYKŁAD



$$\text{Dokładność} = \frac{TP + TN}{TP + FP + TN + FN}$$

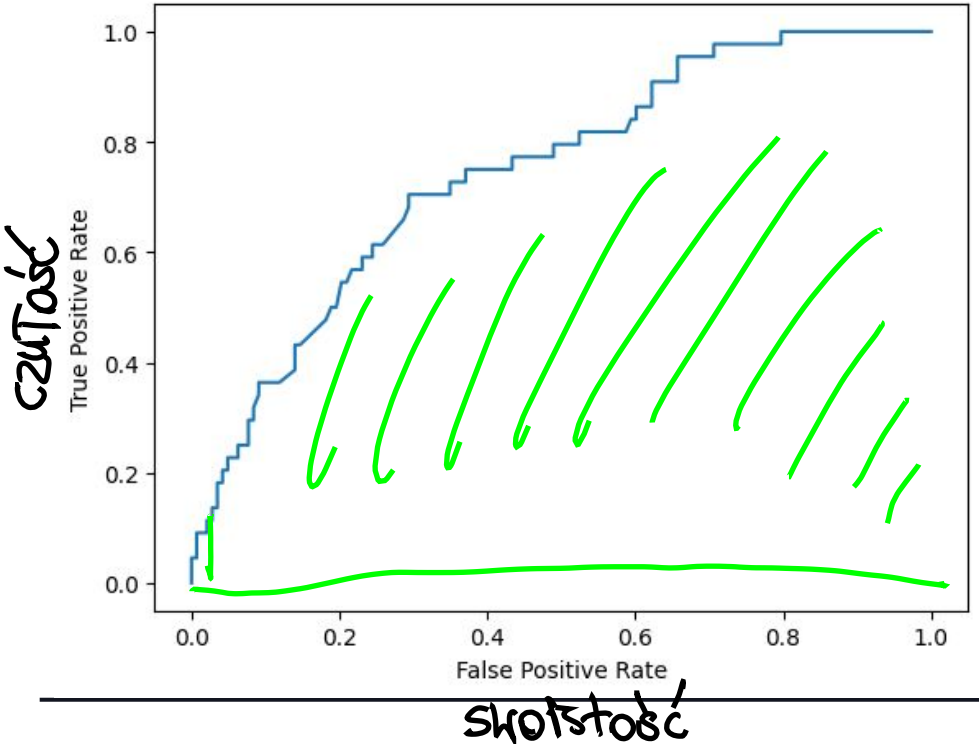
$$\text{Czułość} = \frac{TP}{TP + FN}$$

$$\text{Swoistość} = \frac{TN}{TN + FP}$$

$$\text{Precyzja} = \frac{TP}{TP + FP}$$

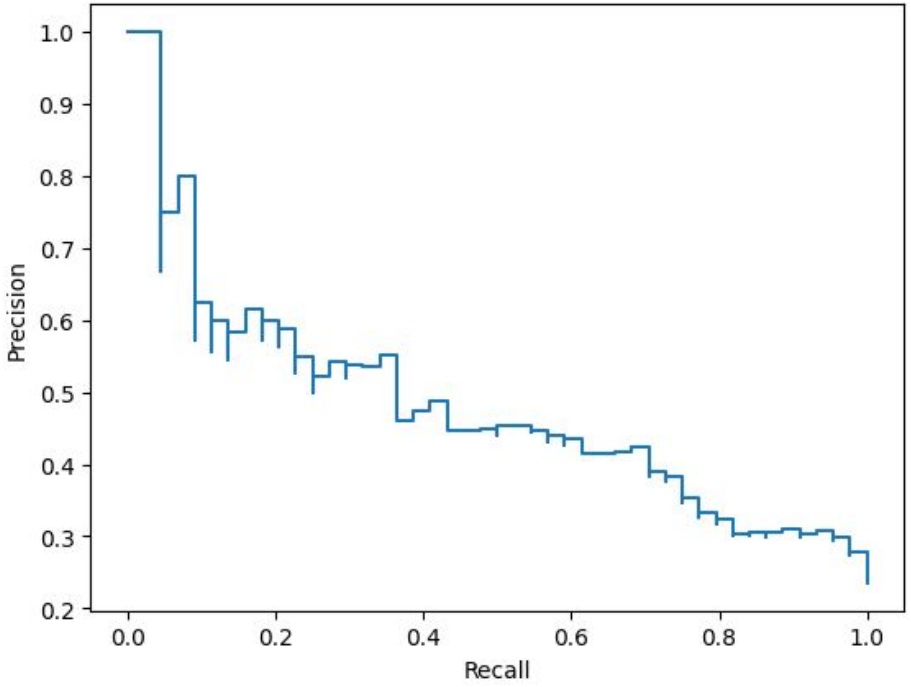
$$\text{Zwrot} = \frac{TP}{TP + FN}$$

KRZYWA ROC



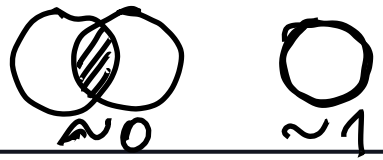
- Krzywa ROC pokazuje kompromis pomiędzy swoistością a czułością klasyfikatora dla różnych progów funkcji decyzyjnej.
- Funkcja decyzyjna klasyfikatora zwraca odległość danego punktu od granic decyzyjnych klasyfikatora.
- Krzywa ROC powstaje poprzez zmianę progu, powyżej którego następuje pozytywna klasyfikacja.
- Im większe pole pod krzywą - tym lepszy klasyfikator.
- Nie każdy klasyfikator posiada zdefiniowaną funkcję decyzyjną np. drzewo decyzyjne nie ma.

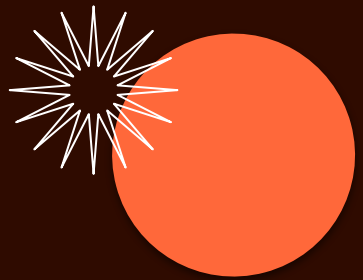
KRZYWA PRECYZJA-ZWROT



- Krzywa precyzji-zwrotu pokazuje kompromis pomiędzy precyzją a zwrotem dla różnych progów funkcji decyzyjnej.
- Im większe pole pod krzywą - tym lepszy klasyfikator.

$$F_1 = 2 \cdot \frac{1 \quad 1}{\frac{1}{\text{PRECISION}} \cdot \frac{1}{\text{RECALL}}}$$





03

REGRESJA LOGISTYCZNA

MODEL LINIOWY

- Model liniowy to suma ważona (kombinacja liniowa) zmiennych, która przewiduje wartość zmiennej tłumaczonej (zależnej, wynikowej) na podstawie wartości zmiennych tłumaczących (niezależnych, wejściowych).
- Przykład: przewidywanie cen domów
- Zmienne:
  - wiek domu w latach X_{AGE}
  - wysokość rocznego podatku od nieruchomości X_{TAX}
- Model:

$$\widehat{Y_{PRICE}} = 212000 + 109X_{TAX} - 2000X_{AGE}$$

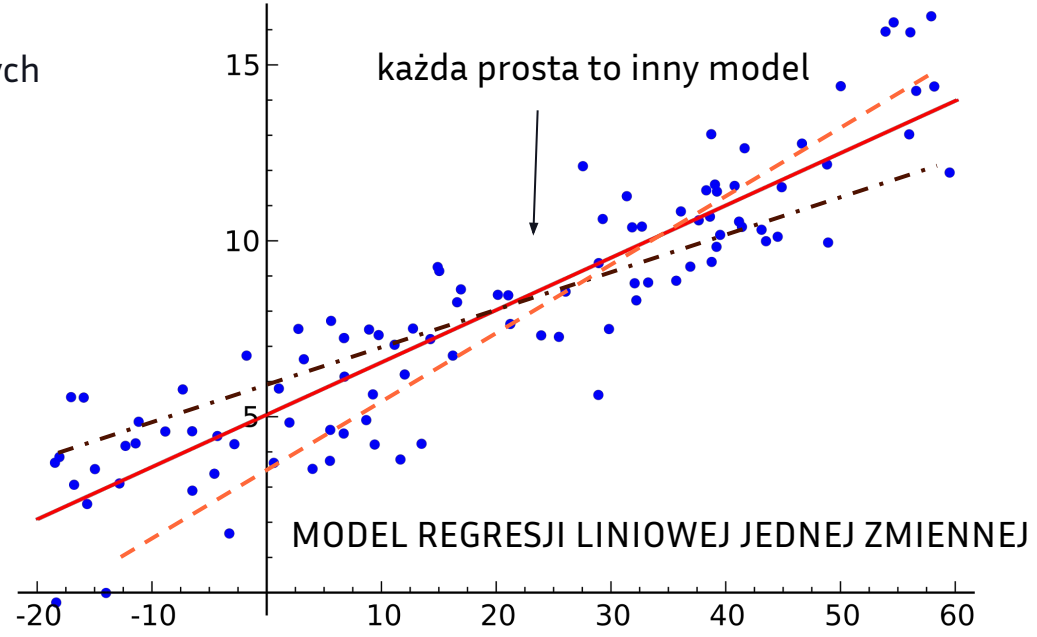
- Przykładowo dla domu opisanego przez krotkę zmiennych tłumaczących
  $(X_{AGE}, X_{TAX}) = (75, 10000)$
  predykowana wartość domu wynosi $\widehat{Y_{PRICE}} = 212000 + 109 \cdot 10000 - 2000 \cdot 75 = 1152000.$

MODEL REGRESJI LINIOWEJ

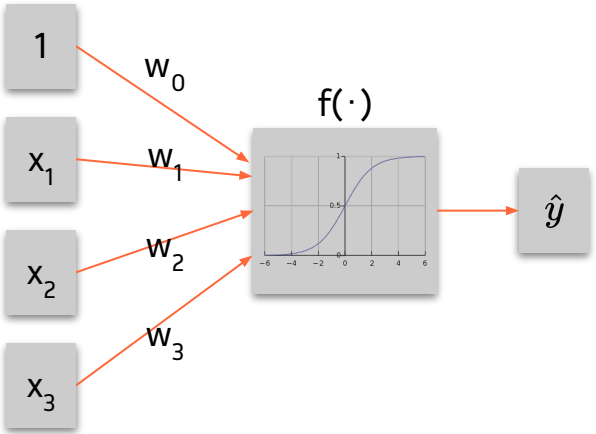
- w - wagi / współczynniki
- x - wektor zmiennych tłumaczących
- w_0 - wyraz wolny, błąd (ang. bias)

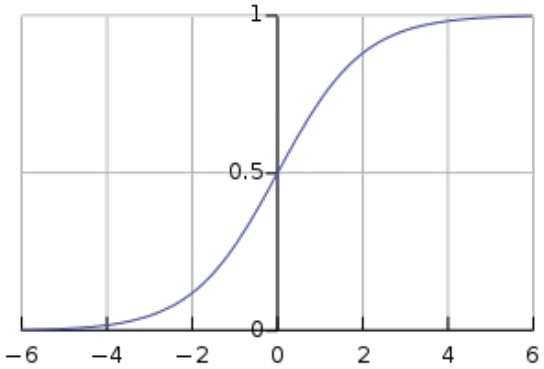
$$\mathbf{x} = (x_1, \ldots, x_n)$$
$$\hat{y} = w_0 + w_1 x_1 + \cdots + w_n x_n$$

$$\mathbf{x} = (x_0, x_1, \ldots, x_n) \text{ gdzie } x_0 = 1$$
$$\hat{y} = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n$$



REGRESJA LOGISTYCZNA





$$\hat{y} = f(w_0 + w_1 x_1 + \cdots + w_n x_n)$$

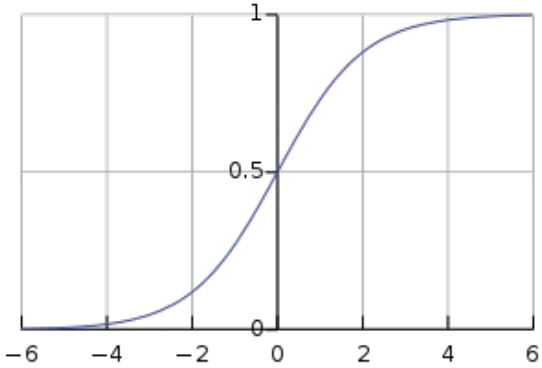
f – funkcja logistyczna

$$f(z) = \frac{1}{1 + \exp(-z)}$$

$z = w_0 + w_1 x_1 + \cdots + w_n x_n$ – wynik regresji liniowej

REGRESJA LOGISTYCZNA

Funkcja logistyczna przekształca zmienną rzeczywistą do wartości pomiędzy 0 i 1, która jest interpretowana jako prawdopodobieństwo, że obiekt wejściowy, dany poprzez wektor zmiennych niezależnych **x**, należy do pozytywnej klasy.



$$\hat{y} = f(w_0 + w_1 x_1 + \cdots + w_n x_n)$$

f – funkcja logistyczna

$$f(z) = \frac{1}{1 + \exp(-z)}$$

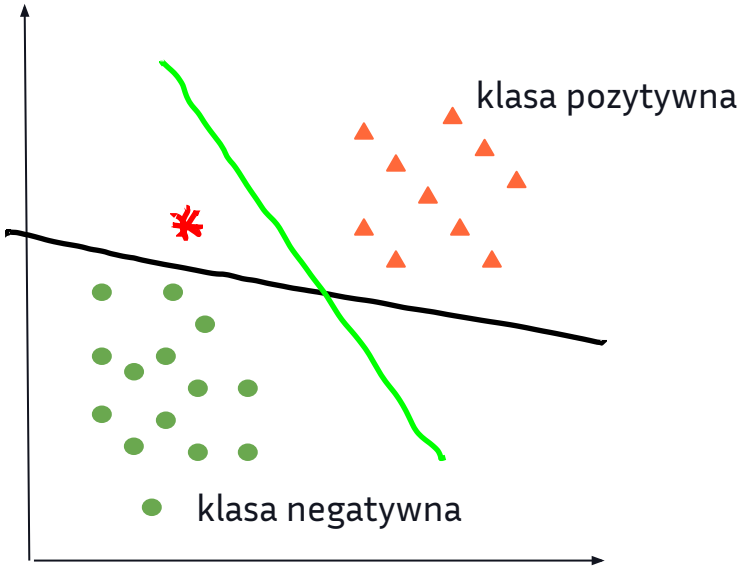
$z = w_0 + w_1 x_1 + \cdots + w_n x_n$ – wynik regresji liniowej

04

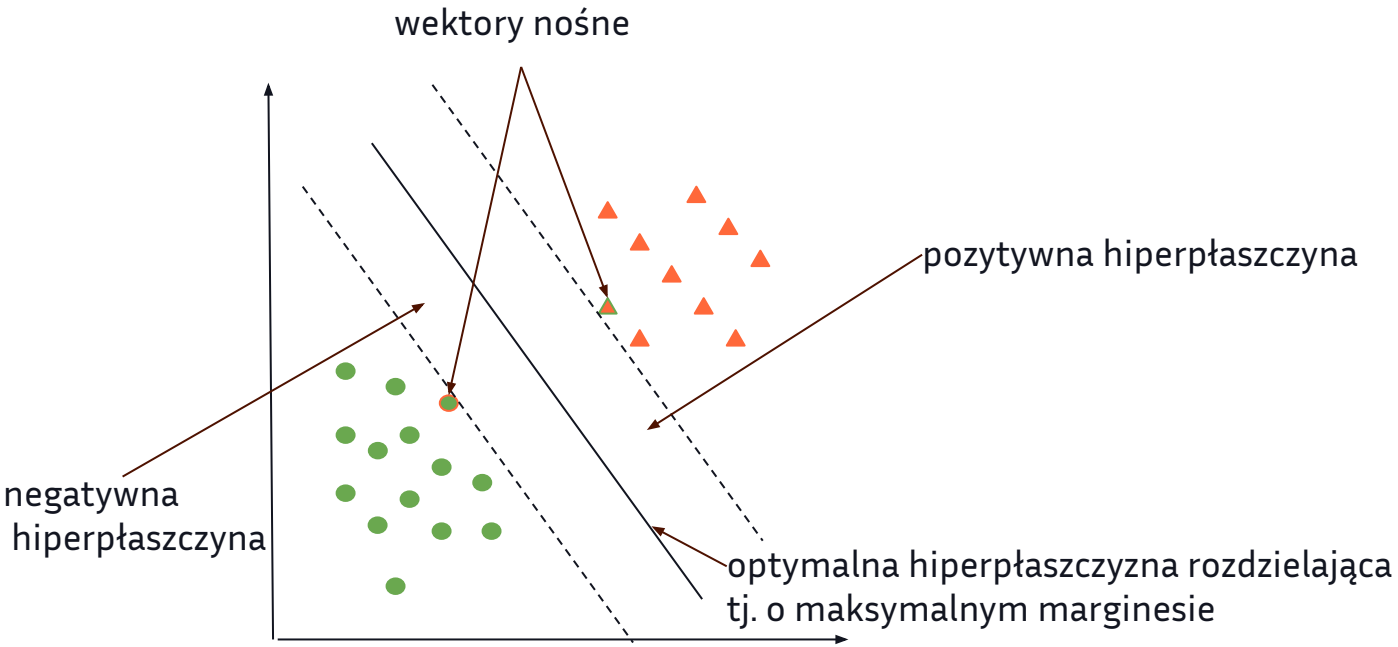
MASZYNY WEKTORÓW NOŚNYCH

MASZYNA WEKTORÓW NOŚNYCH (ang. Support Vector Machine (SVM))

Jak rozdzielić dwie klasy za pomocą jednej linii?



MASZYNA WEKTORÓW NOŚNYCH (ang. Support Vector Machine (SVM))



MASZYNA WEKTORÓW NOŚNYCH

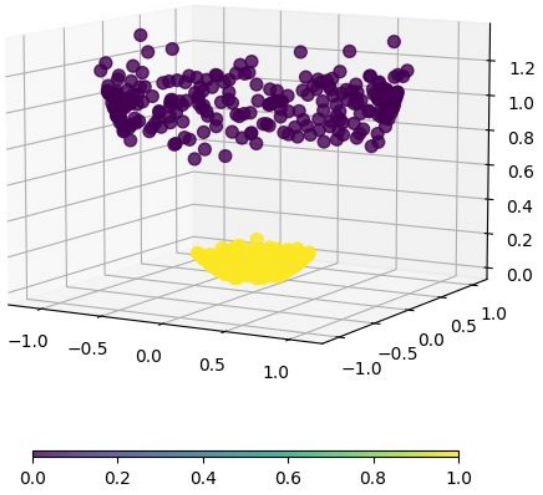
- Marginesem hiperpłaszczyzny rozdzielającej nazywamy odległość tej hiperpłaszczyzny od najbliższego wektora cech próbki w zbiorze uczącym.
- Optymalną hiperpłaszczyzną rozdzielającą OSH (ang. Optimal Separating Hyperplane) nazywamy hiperpłaszczyznę rozdzielającą charakteryzującą się maksymalnym marginesem.
- Wektorami nośnymi (podpierającymi) SV (ang. Support Vector) nazywamy wektory zbioru uczącego położone najbliżej optymalnej hiperpłaszczyzny rozdzielającej.
- Omawiany przykład jest **liniowo separowalny (zbiory są wypukłe)**, gdzie istnienie hiperpłaszczyzny poprawnie rozdzielającej wszystkie próbki zbioru uczącego **jest gwarantowane**.
- W przypadku liniowo nieseparowalnym nie istnieje hiperpłaszczyzna rozdzielająca zapewniająca poprawną klasyfikację wszystkich elementów zbioru uczącego. W takim przypadku poszukujemy hiperpłaszczyzny, która minimalizuje prawdopodobieństwo błędnej klasyfikacji poprzez wprowadzenie tak zwanego *miękkiego marginesu*.

A co w przypadku nieliniowym?

REPREZENTACJA STRUKTURY NIELINIOWEJ (WYKŁAD 6)

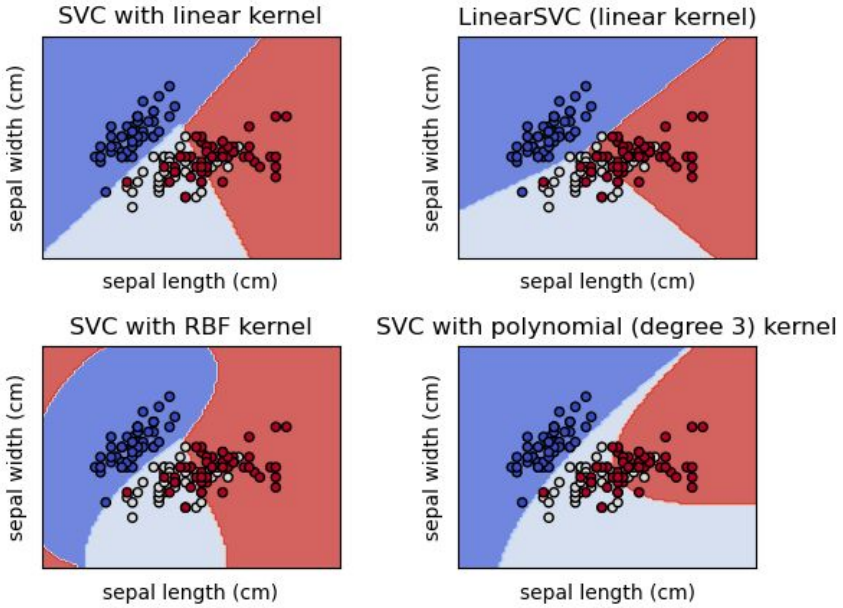
- Rzutowanie do wyższego wymiaru może uprościć dane, których nie da się oddzielić liniowo.
- Zauważmy, że chociaż N punktów nie może być ogólnie rozdzielonych liniowo w d<N wymiarach, prawie zawsze można je liniowo rozdzielić w d>=N wymiarach.

$$(x_1, x_2) \Rightarrow (x_1, x_2, x_1^2 + x_2^2)$$



NIELINIOWA MASZYNA WEKTORÓW NOŚNYCH

- Powierzchnie rozdzielające klasy w przypadku większości rzeczywistych zbiorów danych mają charakter nieliniowy.
- Rozwiązanie: nieliniowa transformacja zbioru wektorów wejściowych do przestrzeni o wyższym wymiarze niż przestrzeń wejściowa.



Źródło grafiki: https://scikit-learn.org/stable/auto_examples/svm/plot_iris_svc.html#sphx-glr-auto-examples-svm-plot-iris-svc-py

FUNKCJA JĄDRA

Jądro wielomianowe

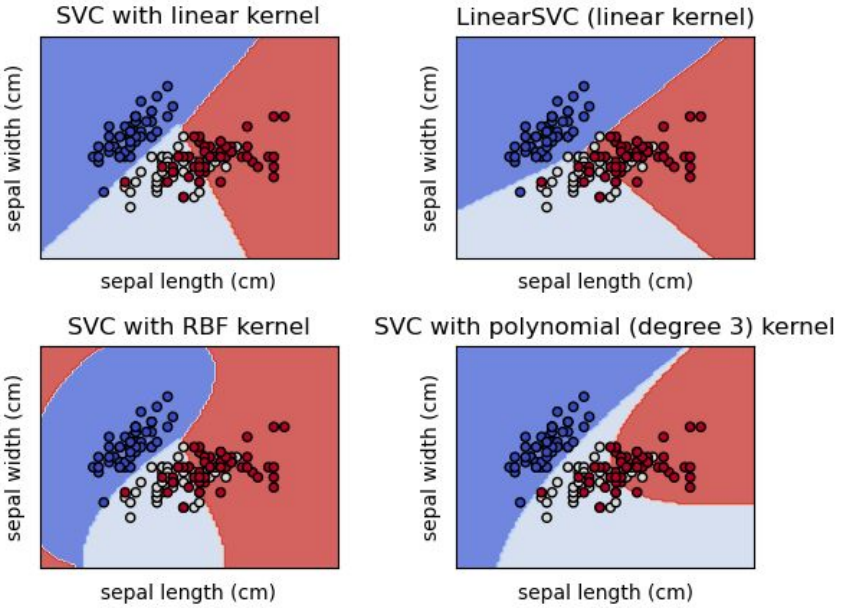
$$K(x_i, x_j) = (x_i^T x_j + b)^p$$

Jądro Gaussowskie (RBF)

$$K(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\right)$$

Jądro liniowe

$$K(x_i, x_j) = x_i^T x_j$$



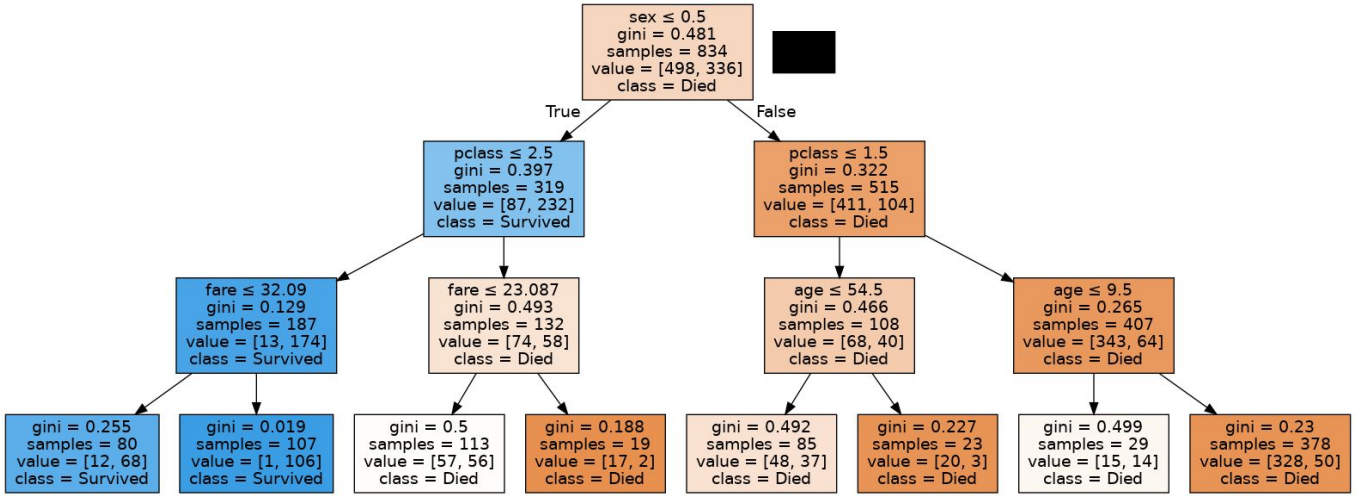
05

DRZEWA DECYZYJNE

DRZEWO

Drzewo – graf nieskierowany, który jest acykliczny i spójny, czyli taki graf, że:

- z każdego wierzchołka drzewa można dotrzeć do każdego innego wierzchołka (spójność)
- i tylko jednym sposobem (acykliczność, brak możliwości chodzenia „w kółko”).



DRZEWO DECYZYJNE (DECISION TREE)

Algorytm działa rekurencyjnie dla każdego węzła drzewa.

Musimy podjąć decyzję, czy węzeł będzie:

1. liściem według kryterium stopu – kończymy to wywołanie rekurencyjne
2. węzłem rozgałęziającym się według kryterium wyboru atrybutu – dokonujemy wyboru atrybutu, tworzymy rozgałęzienia według wartości, jakie przyjmuje dany atrybut, i dla każdego węzła potomnego tworzymy rekurencyjne wywołanie algorytmu, z listą atrybutów zmniejszoną o właśnie wybrany atrybut.

Wszystkie algorytmy działają według podanego schematu, różnice w implementacji dotyczą kryteriów stopu i wyboru atrybutu.

06

SIECI NEURONOWE

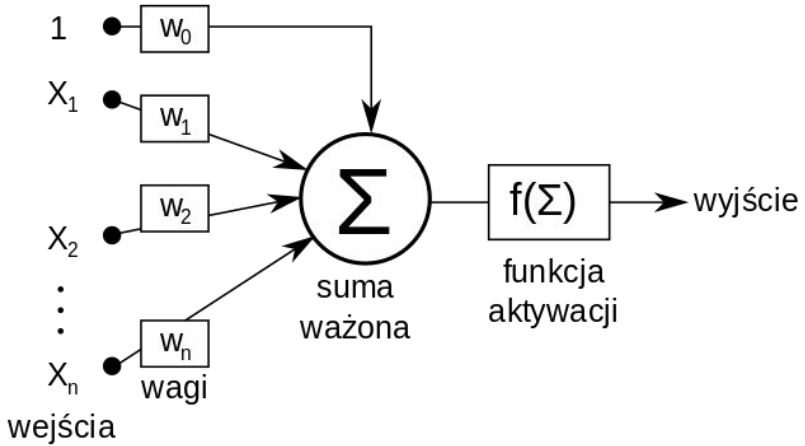
NEURON McCullocha-Pittsa

Wartość na wyjściu neuronu obliczana jest w następujący sposób:

1. obliczana jest suma iloczynów wartości x_i podanych na wejścia i wag w_i wejść:

$$s = w_0 + \sum_{i=1}^{n} x_i w_i$$

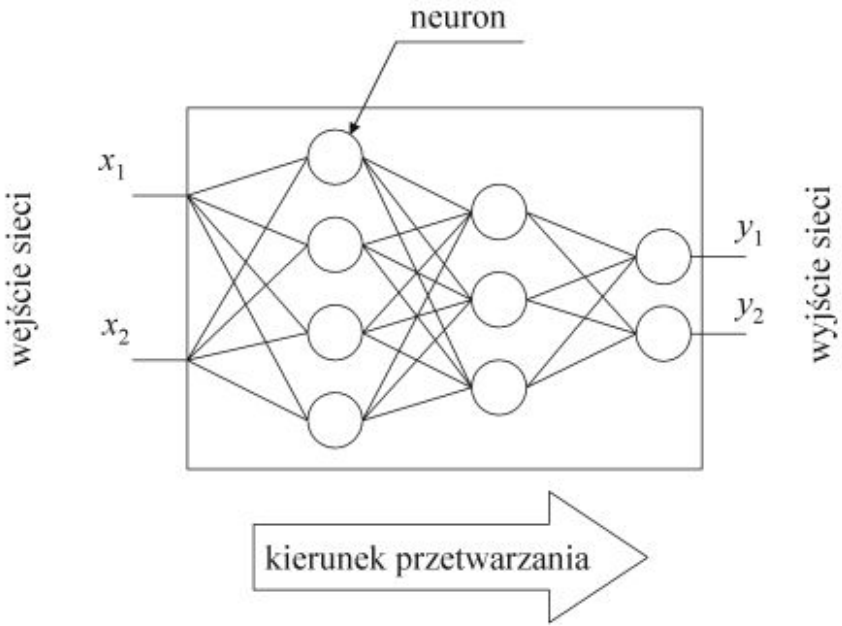
2. na wyjście podawana jest wartość funkcji aktywacji f(s) dla obliczonej sumy



PERCEPTRON

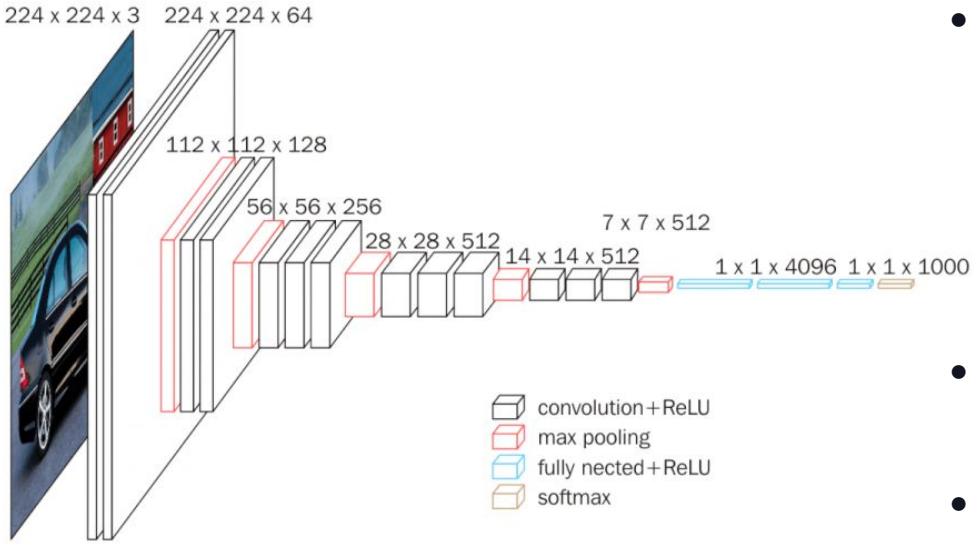
- Najprostsza sieć neuronowa, składająca się z jednego bądź wielu niezależnych neuronów McCullocha-Pittsa.
- Perceptron potrafi określić przynależność parametrów wejściowych do jednej z dwóch klas, poprzez wskazanie czy coś należy czy nie do pierwszej klasy.
- Może być wykorzystywany tylko do klasyfikowania zbiorów liniowo separowalnych.
- Aby móc testować przynależność do więcej niż dwóch klas, należy użyć perceptronu z większą ilością neuronów.

PERCEPTRON WIELOWARSTWOWY



Źródło grafiki: https://wazniak.mimuw.edu.pl/images/e/e3/SI_M12_01.png

KONWOLUCYJNE SIECI NEURONOWE



https://paperswithcode.com/sota/image-classification-on-imagenet

- VGG16 to konwolucyjny model sieci neuronowej zaproponowany przez K. Simonyana i A. Zissermana z Uniwersytetu Oksfordzkiego w artykule „Very Deep Convolutional Networks for Large-Scale Image Recognition”.
- Model osiąga 92.7% top-5 dokładności na (pod)zbiorze testowym ImageNet.
- ImageNet jest zbiorem danych zawierającym ponad 14 milionów obrazów należących do 1000 klas.

BONUS

https://lazypredict.readthedocs.io/en/latest/usage.html

THANKS!

DZIĘKUJĘ ZA UWAGĘ