Machine Learning

Wprowadzenie z przykładami w Python

O autorach

Adrianna Napiórkowska

Absolwentka MIESI (SGH) i członkini koła naukowego Business Analytics. Obecnie studiuje Analizę Danych (NOVA IMS)

napiorkowska.adrianna@gmail.com

Jacek Dziwisz

Student Data Science & Big Data Analytics (SGH) i członek koła naukowego Business Analytics.

kontakt@jacekdziwisz.pl

Plan warsztatu

1. Czym jest uczenie maszynowe
2. Proces uczenia maszynowego
3. Typy zadań w jakich wykorzystywane jest uczenie maszynowe
4. Ocena jakości modeli
5. Omówienie kilku modeli z przykładami użycia w Python
    1. Perceptron
    2. Regresja logistyczna
    3. Drzewo decyzyjne
    4. Algorytm k-najbliższych sąsiadów

Machine Learning

„A computer program is said to learn from experience E with respect to som class of task T and performance measure P, if its performance at task in T, as measured by P, improves with experience E.”

(Mitchell 1996)

Uczenie maszynowe

- Obserwacje używane są jako przykłady i służą do budowania wiedzy, która jest bardziej uniwersalna
- Mimika ludzkiego procesu uczenia, czyli uczenie „na przykładach”
  - Parkowanie samochodu
  - Jazda na rowerze
- Zastosowania:
  - Bardzo duża liczba zmiennych objaśniających
  - Zapotrzebowanie na szybkie decyzje
  - Konieczność podjęcia decyzji bezstronnej, bez udziału człowieka

Definicja problemu

Mając dany zbiór **D**, składający się z par: (**x**, y) znaleźć jak najlepsze przybliżenie funkcji (relacji) pomiędzy wektorem **x** a odpowiadającej mu wartością zmiennej objaśnianej y.

$$\forall_{i=1,2,\ldots N} \qquad \varphi(x_i) = y_i$$

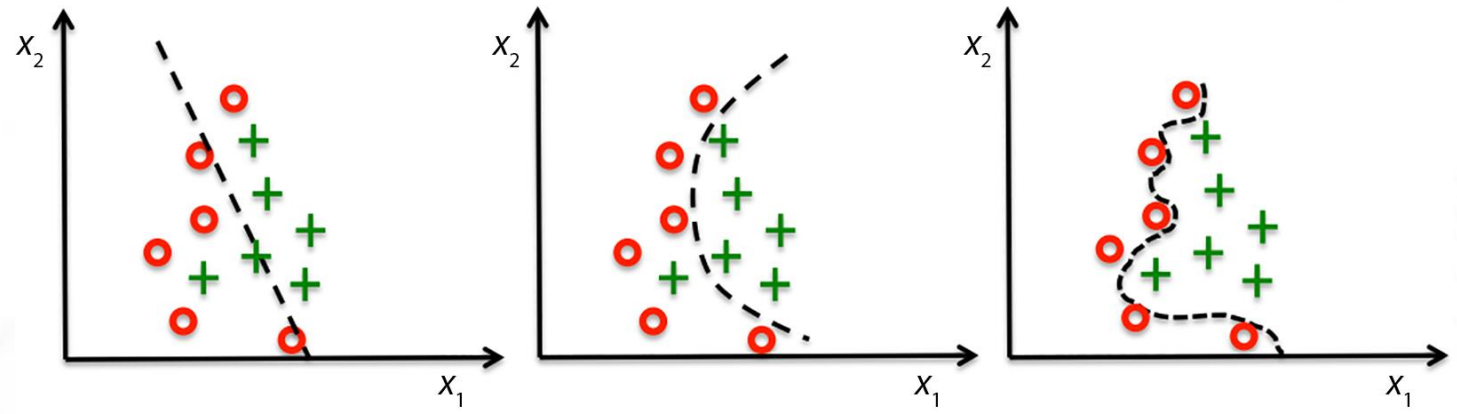
Terminologia:
φ – szukana funkcja
D – zbiór uczący
Uczenie (learning) – proces pozwalający na znalezienie funkcji **f** stanowiącej jak najlepsze przybliżenie funkcji φ
f – model
y_i - wartości zmiennej objaśnianej (target value)

Umiejętność uogólniania

Można powiedzieć, że nasz model **f** ma umiejętność uogólniania, jeśli zachowuję się jak φ również na danych, które nie należą do zbioru uczącego.

Gdy model jest za bardzo dopasowany do danych treningowych, mówimy o **przeuczeniu** (overfitting).

Miarą jakości modeli jest to jak dobrze model klasyfikuje przypadki znajdujące się poza zbiorem uczącym.



Przykład

Zbiór uczący:

x_1	x_2	y
1	8	9
3	3	6
4	2	6
7	3	10

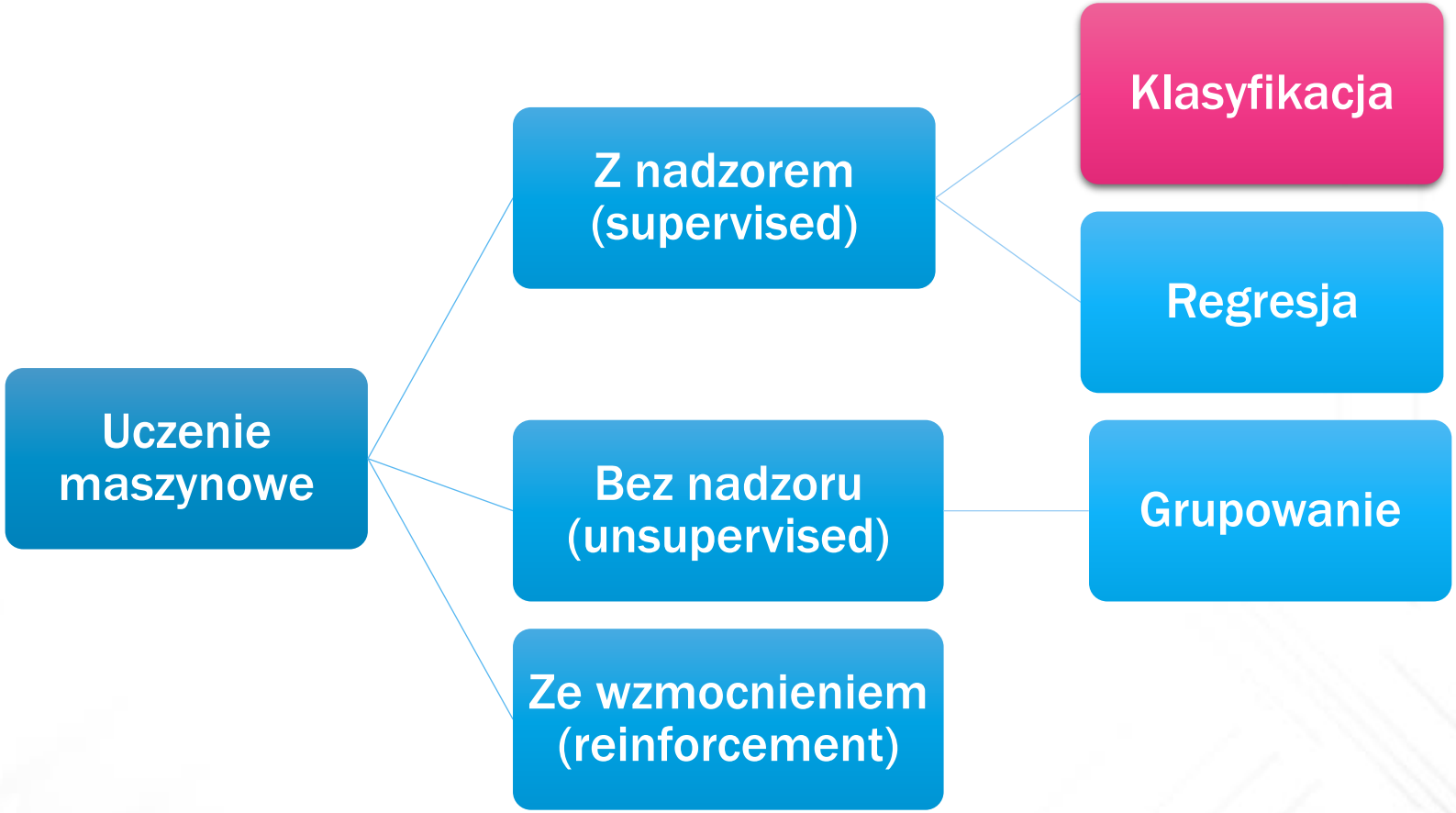
Model 1

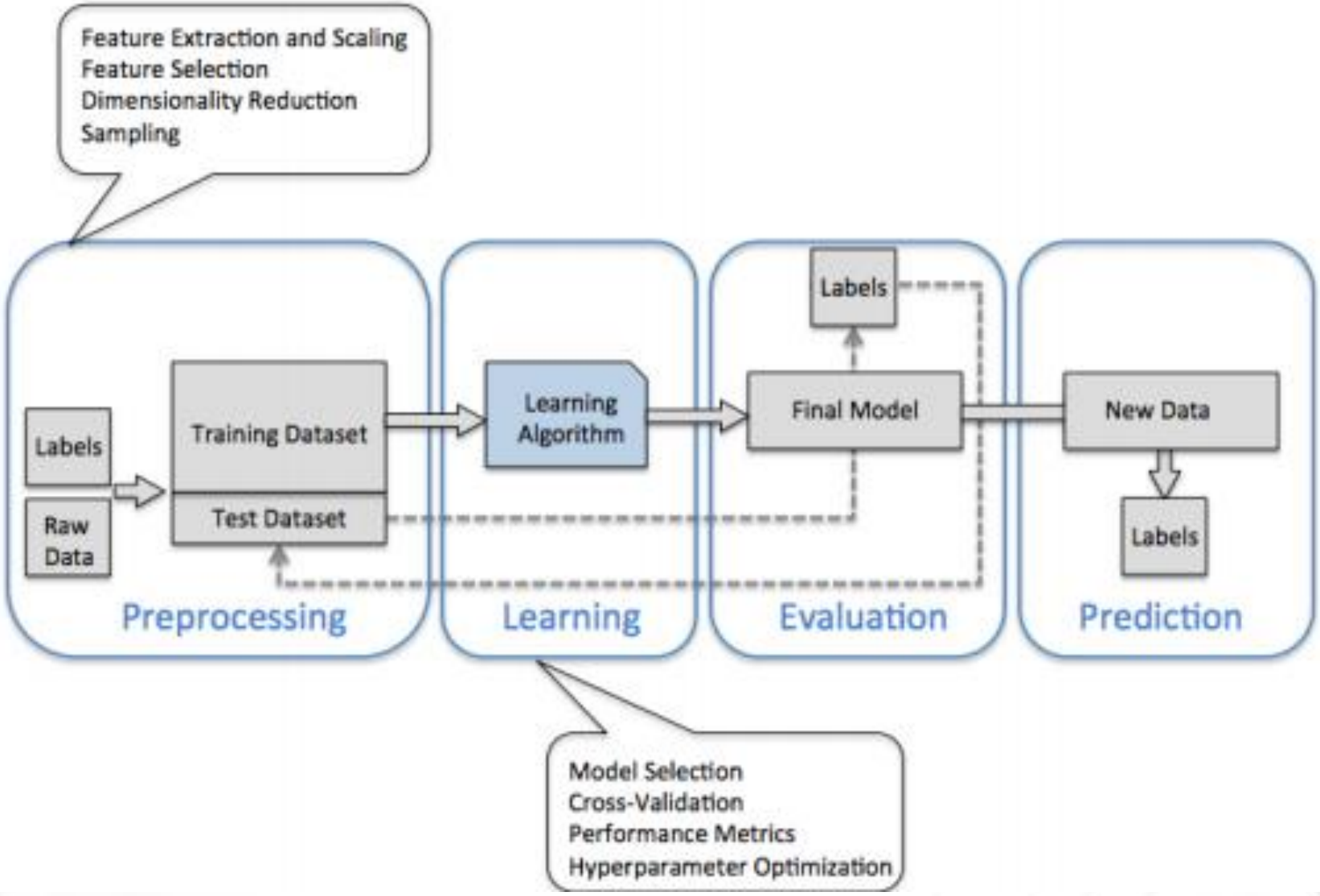
$$\varphi_1(x_1, x_2) = x_1 + x_2$$

Model 2

$$\varphi_2(x_1, x_2) = \text{if } (x_1 == 1) \,\&\, (x_1 == 8)\ return\ 9$$
$$else$$
$$\text{if } (x_1 == 3) \,\&\, (x_1 == 3)\ return\ 6$$
$$else$$
$$\text{if } (x_1 == 4) \,\&\, (x_1 == 2)\ return\ 6$$
$$else$$
$$\text{if } (x_1 == 7) \,\&\, (x_1 == 3)\ return\ 9$$
$$else$$
$$return\ random\ number$$

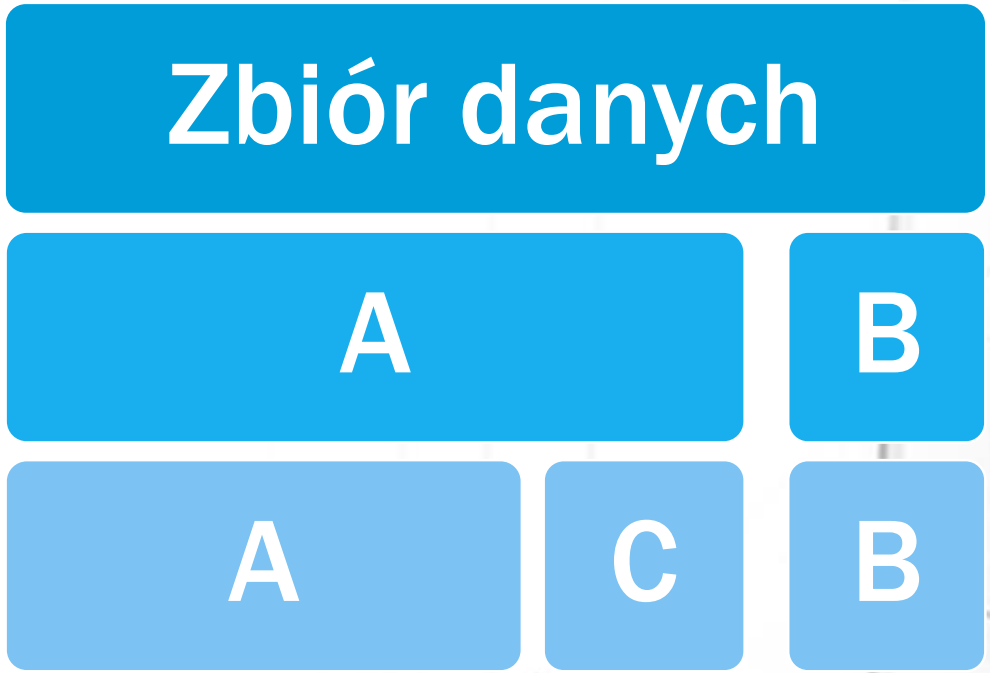
Typy zadań uczenia maszynowego



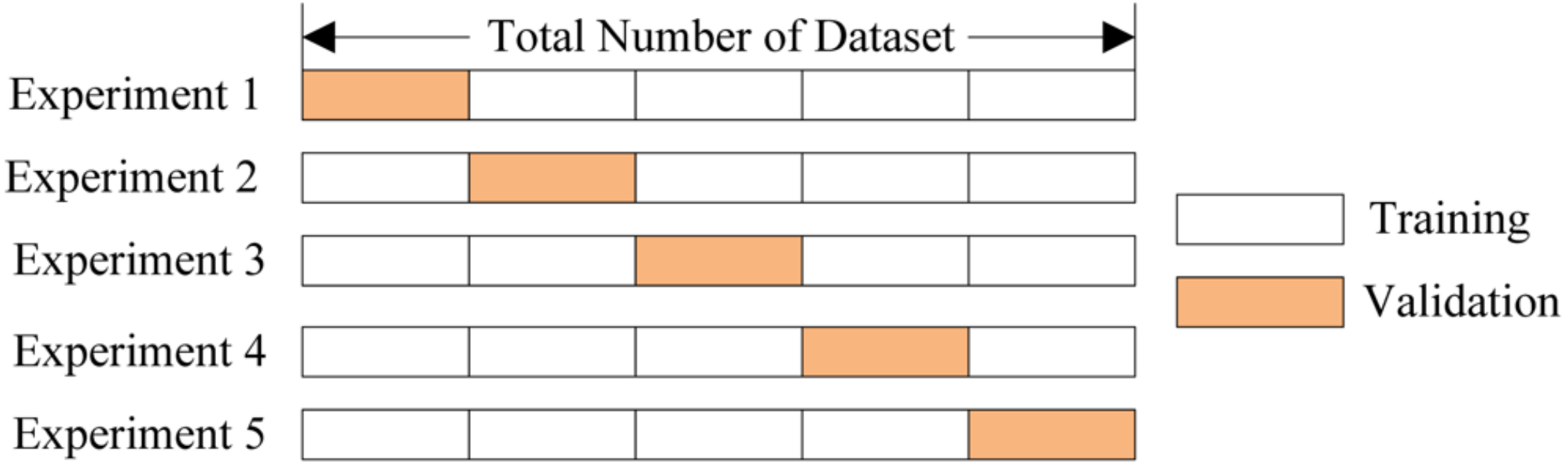

Feature Extraction and Scaling
Feature Selection
Dimensionality Reduction
Sampling
Labels
Raw Data
Training Dataset
Test Dataset
Learning Algorithm
Final Model
Labels
New Data
Labels
Preprocessing
Learning
Evaluation
Prediction
Model Selection
Cross-Validation
Performance Metrics
Hyperparameter Optimization

Ocena jakości modelu – podział danych

- Jak dobrze nasz model klasyfikuje obserwacje nieznajdujące się w zbiorze uczącym
- Aby to sprawdzić należy podzielić zbiór przykładów (obserwacji) na dwie części:
  - Zbiór treningowy (A)
  - Zbiór testowy (B)
- Jeśli chcemy później jeszcze dokładniej porównać modele, możemy wcześniej sprawdzić je na zbiorze walidacyjnym (C), zostawiając zbiór testowy do ostatecznego sprawdzianu



Ocena jakości modelu - walidacja krzyżowa

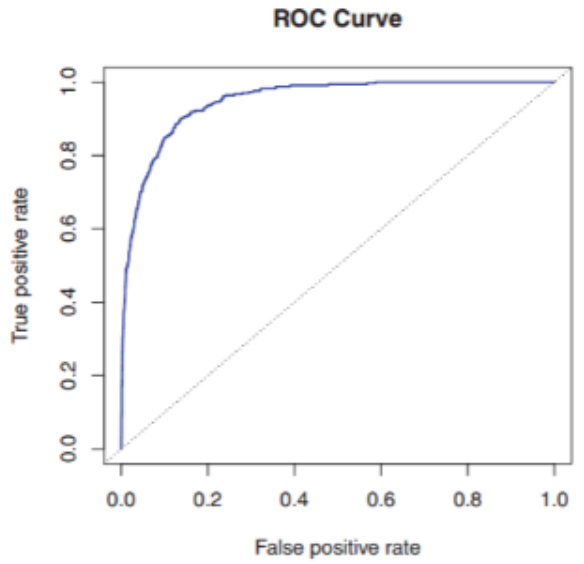


Źródło: https://www.kaggle.com/dansbecker/cross-validation

Ocena jakości modelu

- Tablica trafności
  - Dokładność modelu (accuracy) – (TP+TN)/(TP+FP+FN+TN)
  - Wrażliwość (sensitivity) – TP/(TP+FN)
  - Specyficzność (specificity) - TN/(TN+FP)
- Krzywa ROC (receiver operating characteristic curve)
  - Pole pod krzywą ROC (AUC – area under (ROC) curve)
  - Odsetek poprawnie zaklasyfikowanych obserwacji w zależności od progu decyzji

		Wartości rzeczywiste	
		1	0
Predykcja	1	TP	FP
	0	FN	TN



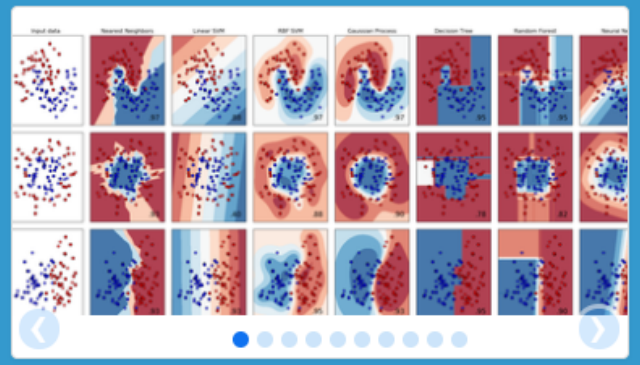
Zbudujmy kilka modeli!

Scikit-learn – uczenie maszynowe w Python

scikit learn

Home | Installation | Documentation | Examples

Google Custom Search | Search



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

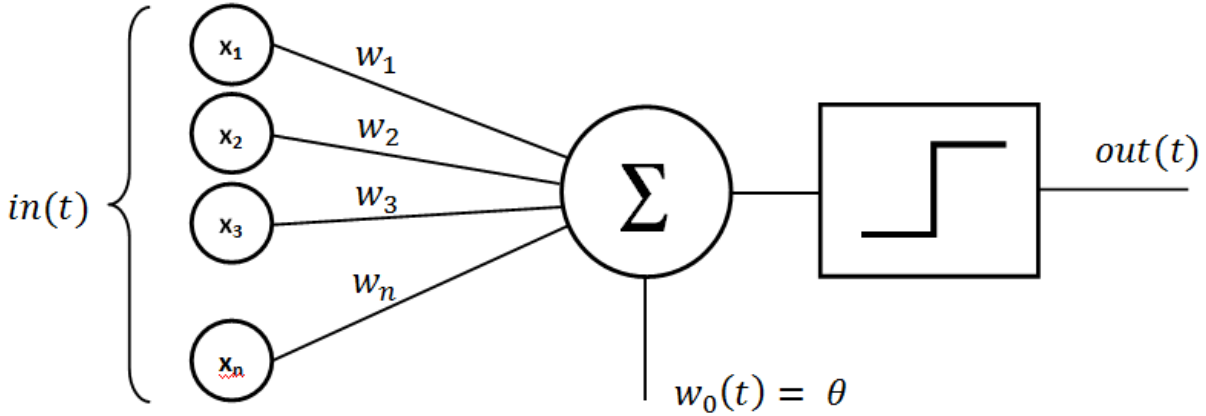
Clustering

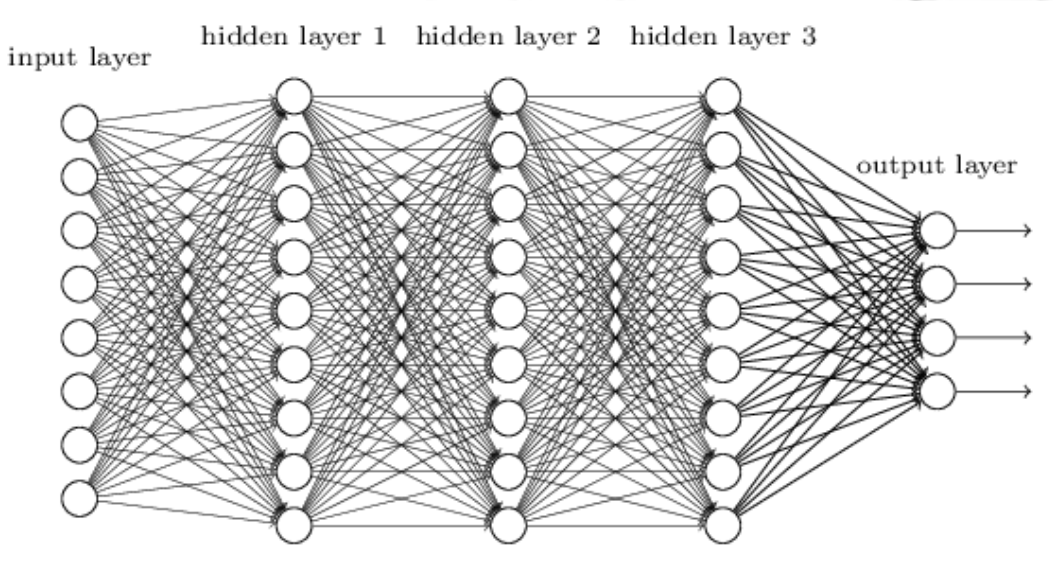
Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Perceptron

- Najprostsza sieć neuronowa
- 3 warstwy:
  - Dane wejściowe – wartości zmiennych objaśniających
  - Neuron agregujący wartości z pierwszej warstwy
    - Wynik przekształcany jest następnie za pomocą funkcji aktywacji w daną wyjściową
  - Dane wyjściowe - predykcja





Źródło: https://pl.wikipedia.org/wiki/Perceptron,
https://ksopyla.com/python/tensorflow/wielowarstwowa-siec-neuronowa-w-tensorflow-do-klasyfikacji-cyfr-z-mnist/

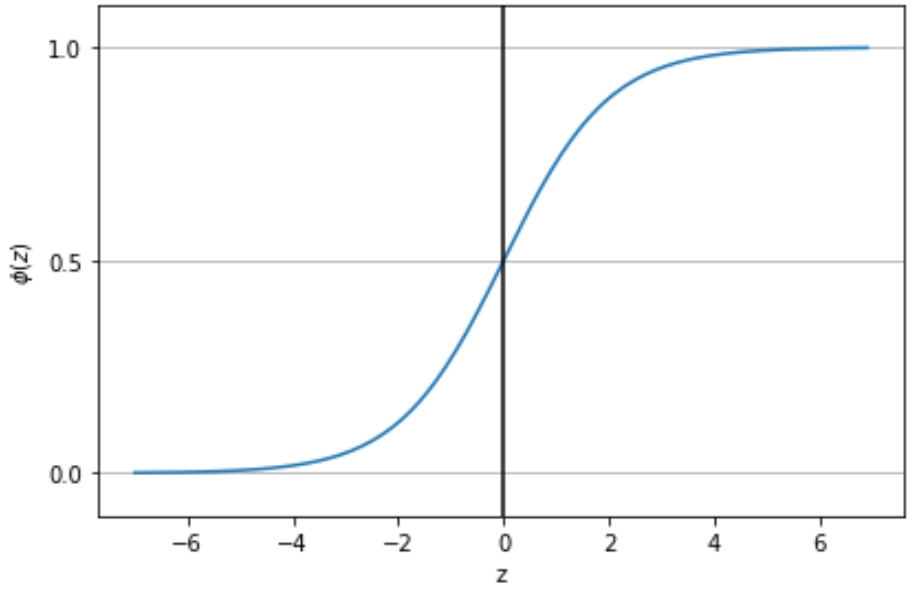
Proces uczenia

- Optymalizacja funkcji kosztu przez odpowiedni wybór wag
- Algorytm:
  1. Zapoczątkuj wektor wag losowymi, małymi liczbami (mogą być zera)
  2. Wybierz szybkość uczenia (learning rate)
  3. Dla każdego przykładu:

     3.1 Oblicz wartość zmiennej objaśnianej wynikającej z modelu

     3.2 Zaktualizuj wagi według wzoru: $w_j := w_j + \Delta w_j$, gdzie $\Delta w_j = \eta(y^{(i)} - \widehat{y^{(i)}})x_j^{(i)}$, a η to wybrana w poprzednim kroku

Regresja logistyczna

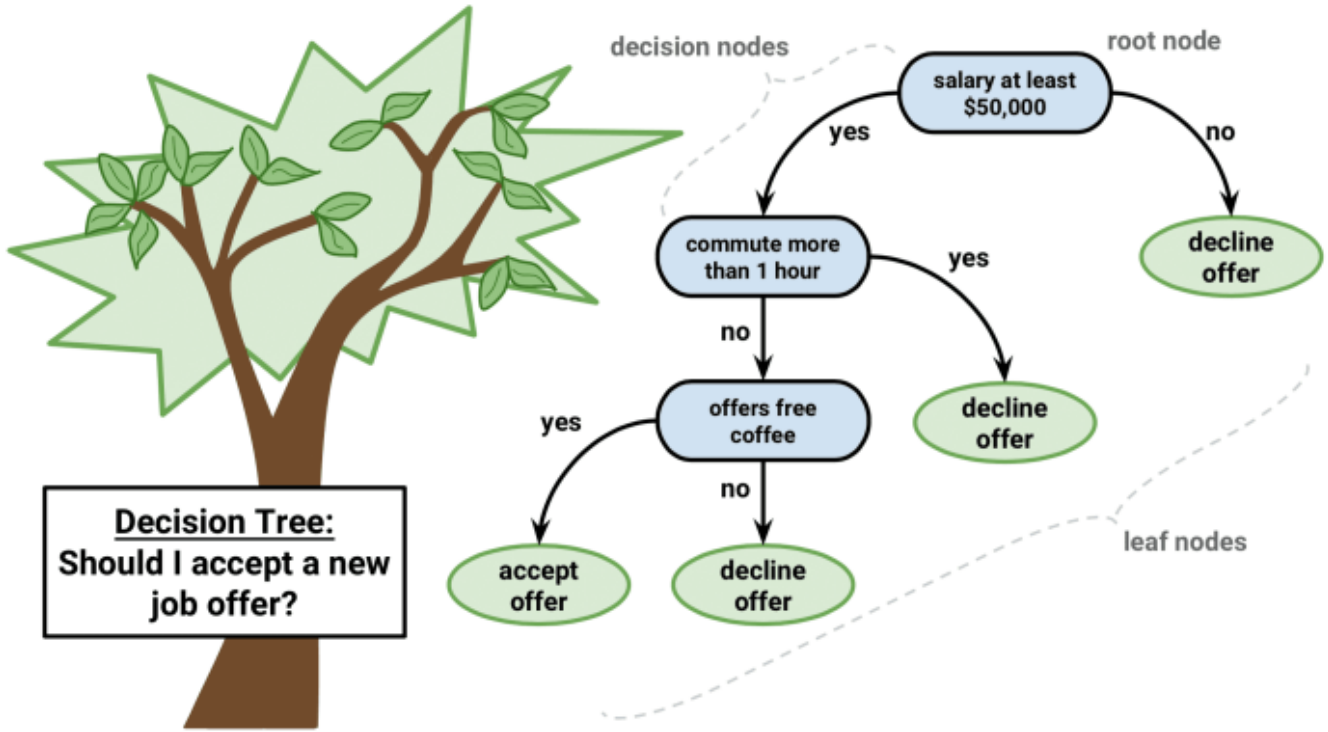


- Funkcja o kształcie litery S
- Iloraz szans (ang. odds ratio, OR) – stosunek szans wystąpienia danego zdarzenia do szans jego niewystąpienia
- Funkcja logitowa $\operatorname{logit}(p) = \ln \frac{p}{1-p}$
- Przekształcenie odwrotne

$$p = \frac{e^{\operatorname{logit}(p)}}{1 + e^{\operatorname{logit}(p)}} = \frac{1}{1 + e^{-\operatorname{logit}(p)}}$$

Drzewa decyzyjne

- Metoda nieparametryczna
- Model bardzo elastyczny i łatwy w interpretacji, lecz często prowadzi do przeuczenia
- Przycinanie drzew
  - Aby uniknąć problemu przeuczenia modelu, jako jeden z parametrów podajemy maksymalną głębokość drzewa



Źródło: https://towardsdatascience.com/decision-tree-hugging-b8851f853486

Miary podziału w drzewach decyzyjnych

- Miary ilości informacji niesionej przez podział zbioru względem danego atrybutu
- Entropia

$$-\sum_{i=1}^{c} p(i|t)\log_2 p(i|t)$$

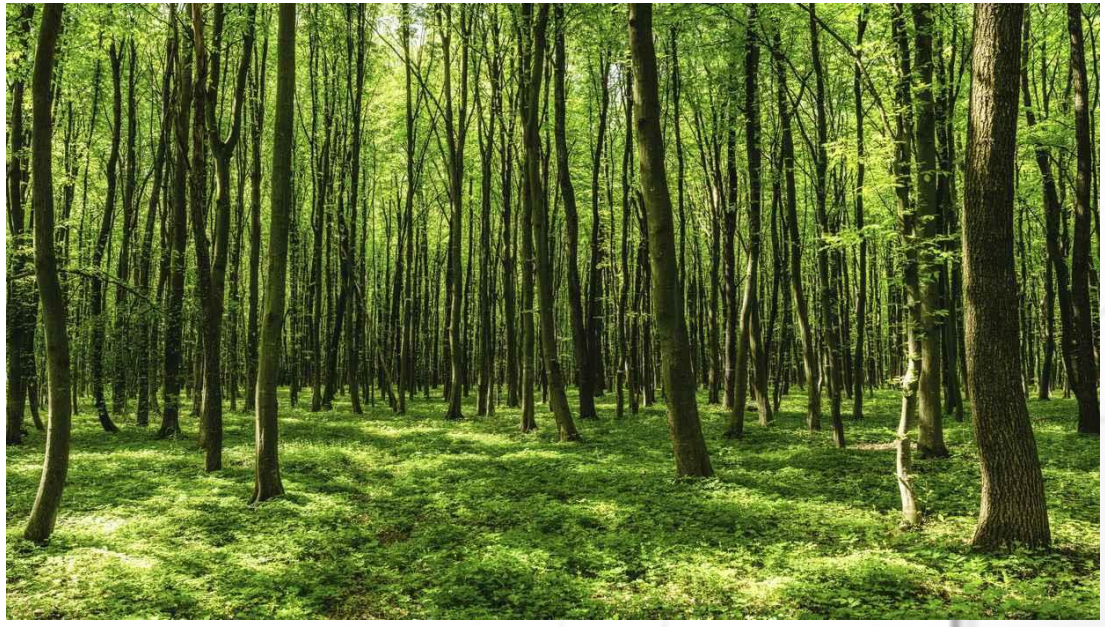
- Współczynnik Ginniego

$$\sum_{i=1}^{c} p(i|t)\left(1-p(i|t)\right)$$

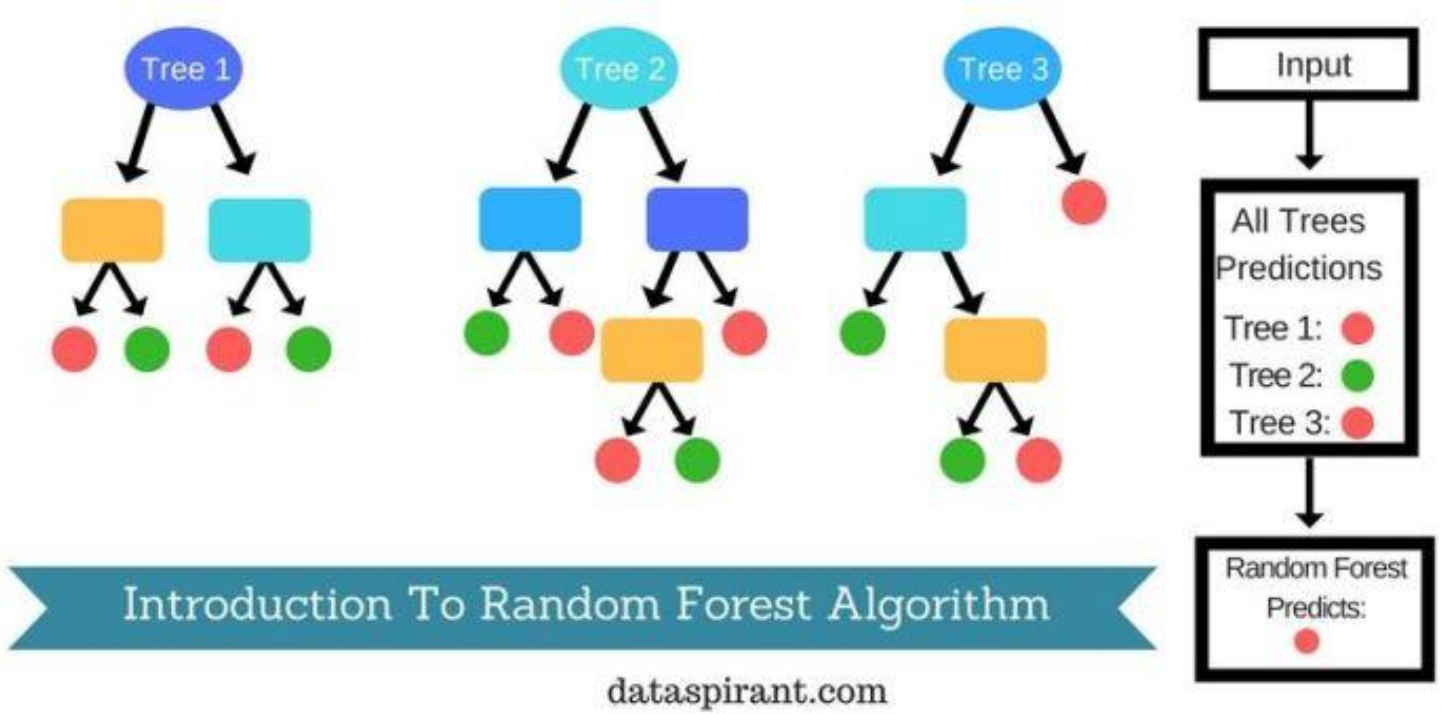
- Maksymalna wartość w sytuacji gdy klasy w liściu są tak samo liczne

Las losowy

- Bagging – uśrednianie wyniku pochodzącego z wielu modeli
- Każde drzewo w lesie losowym budowane jest na losowym podzbiorze zmiennych i obserwacji
- Wiele słabych klasyfikatorów
- Możliwość odczytania istotności zmiennych



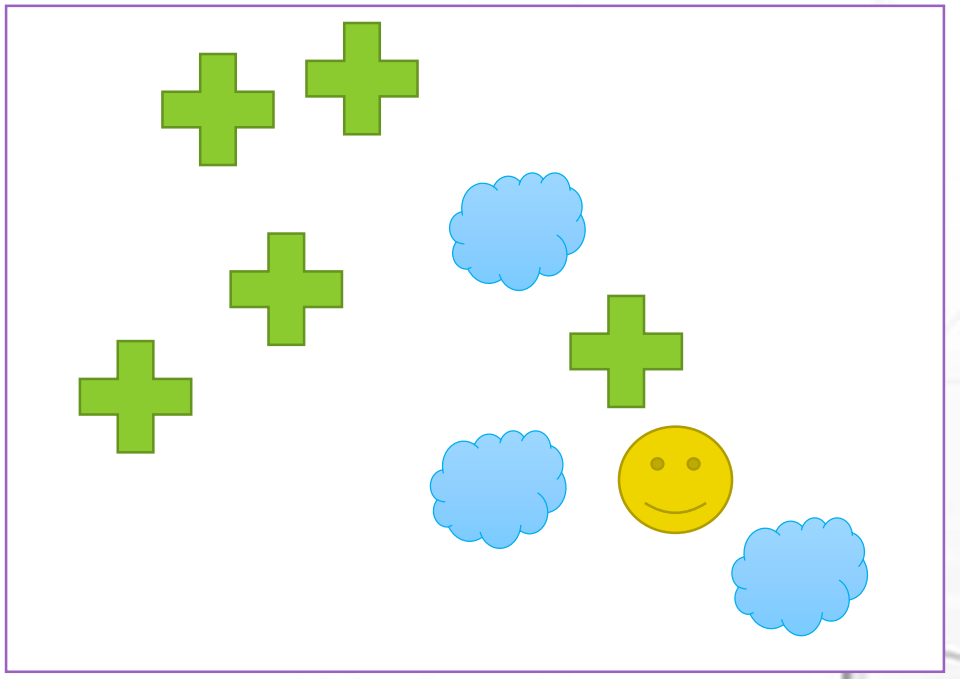
Las losowy



Źródło: http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/

Algorytm k–najbliższych sąsiadów

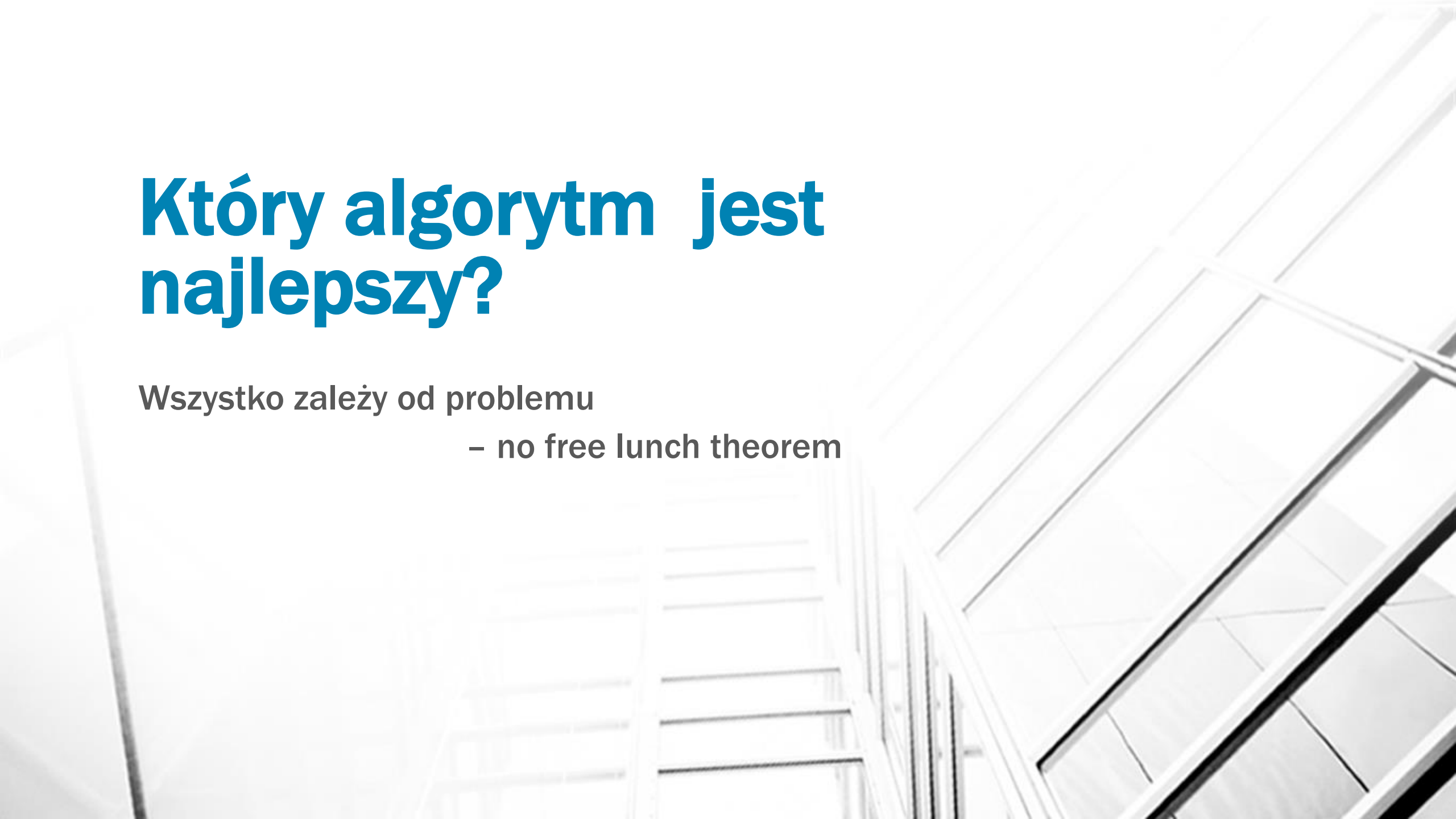
- Metoda nieparametryczna
- Uczenie przez zapamiętywanie wszystkich obserwacji
- Wybór k punktów znajdujących się najbliżej obserwacji, którą mamy zaklasyfikować
- Ustalenie klasy poprzez głosowanie większościowe



Który algorytm jest najlepszy?

Wszystko zależy od problemu

– no free lunch theorem



Pytania?

napiorkowska.adrianna@gmail.com
kontakt@jacekdziwisz.pl

Dziękujemy za uwagę!

Bibliografia

- S. Raschka, *Python Machine Learning*
- T. Mitchell, *Machine Learning*
- T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*
- scikit-learn.org