Wstęp do Machine Learningu

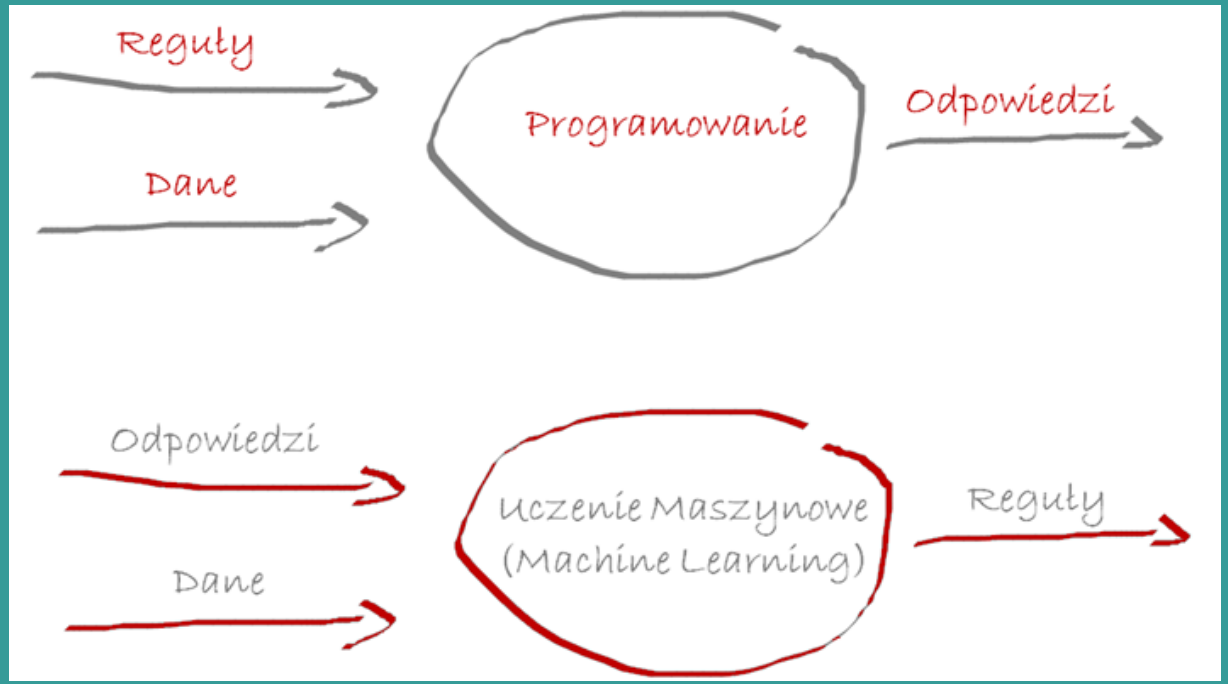
Cześć

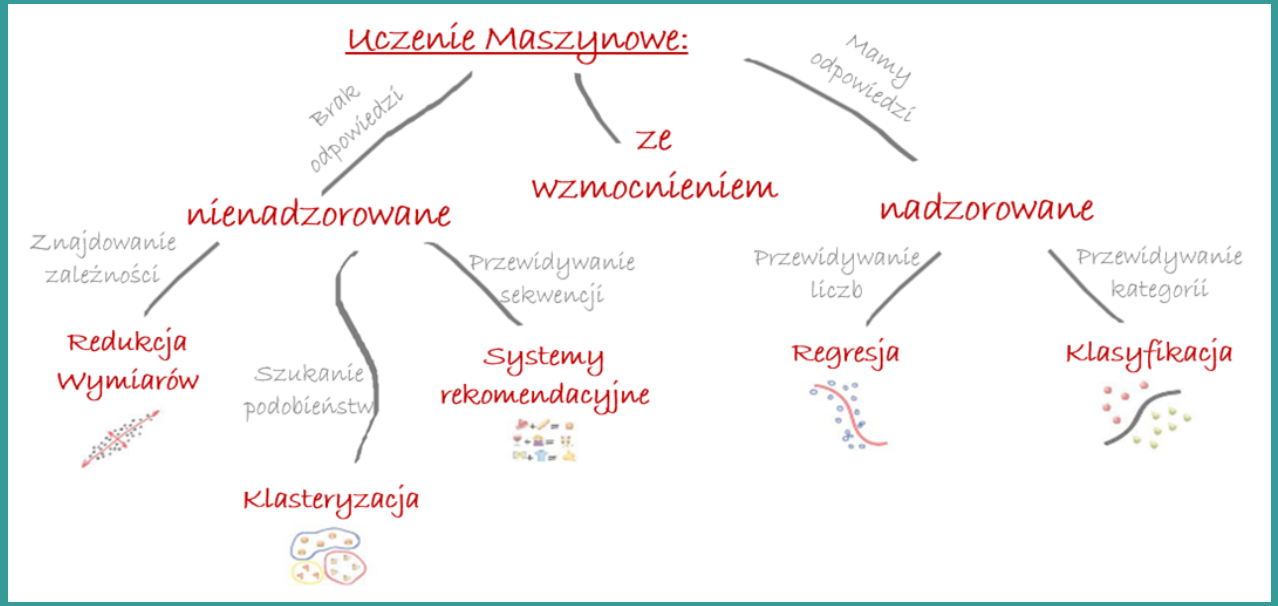
Roman Trinczek

Plan na dziś

1.Co to jest i jakie problemy może rozwiązać ML?
2.Jak mierzyć wydajność naszych modeli?
3.Co to jest “overfitting” czyli nadmierne dopasowanie?
4.Jak działa “feature engineering” i na czym polega optymalizacja parametrów modelu?

Co to jest Machine Learning?


Reguły
Dane
Programowanie
Odpowiedzi
Odpowiedzi
Dane
Uczenie Maszynowe
(Machine Learning)
Reguły


Uczenie Maszynowe:
Brak odpowiedzi
ze wzmocnieniem
Mamy odpowiedzi
nienadzorowane
nadzorowane
Znajdowanie zależności
Przewidywanie sekwencji
Przewidywanie liczb
Przewidywanie kategorii
Redukcja Wymiarów
Szukanie podobieństw
Systemy rekomendacyjne
Regresja
Klasyfikacja
Klasteryzacja

1.1 Uczenie nadzorowane

Uczenie nadzorowane

W uczeniu nadzorowanym (ang. Supervised Learning) model ma dostęp zarówno do zmiennych wejściowych (**X**) jak i zmiennej wyjściowej (**Y**). Celem modelu jest nauczenie się jak przekształcić wejście na wyjście:

$$Y=f(X)$$

Regresja vs Klasyfikacja

- Regresja - zmienna wyjściowa **Y** jest ciągła, inaczej mówiąc jest liczbą rzeczywistą. Przykładowo pensja mierzona w PLN, długość mierzona w centymetrach.

- Klasyfikacja - zmienna wyjściowa **Y** jest dyskretna, inaczej mówiąc jest kategorią. Na przykład przewidujemy kolory: “czarny”, “czerwony”... lub przewidujemy czy dana nieruchomość jest atrakcyjna: “warta zakupu” lub “niewarta zakupu”.

Regresja: Co może być wejściem a co wyjściem?

Problem

Przewidywanie zarobków.

Wejście (X)

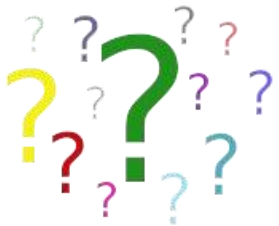
1. Kierunek studiów
2. Średnia ocen
3. Znajomość języków obcych
4. Staż pracy.
5. ???

Wyjście (Y)

Kwota rocznych zarobków brutto (wyrażona w PLN).

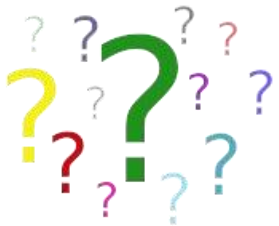
Regresja: Co może być wejściem a co wyjściem?

Problem



Wejście

Wyjście



Klasyfikacja: Co może być wejściem a co wyjściem?

Problem	Wejście	Wyjście
Rozpoznawanie czy na obrazie występuje kot.	1. Zdjęcie	Odpowiedź: tak lub nie.

Klasyfikacja: Co może być wejściem a co wyjściem?

Problem

Wejście

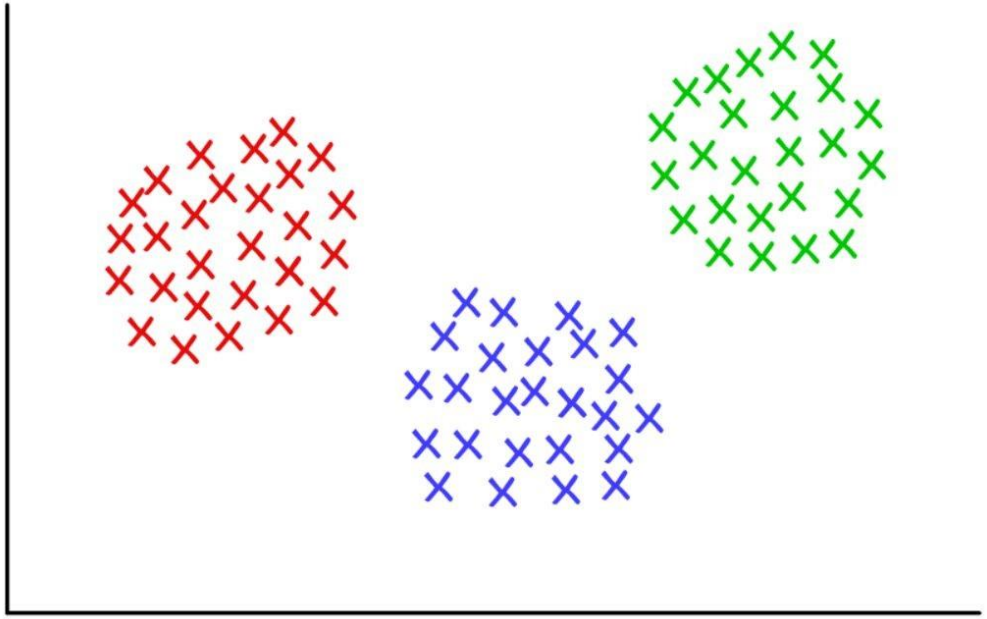
Wyjście

12Uczenie nienadzorowane

Uczenie nienadzorowane

W uczeniu nienadzorowanym (ang. Unsupervised Learning) nasz model ma dostęp tylko do zmiennych wejściowych (**X**) i brak jest zmiennej **Y.** Nie ma czegoś takiego jak poprawna odpowiedź, a celem modelu jest nauka wewnętrznej struktury danych.

Uczenie nienadzorowane - klastrowanie



13Uczenie ze wzmocnieniem

Uczenie ze wzmocnieniem (Reinforcement Learning)

2. Inżynieria cech

Feature Engineering

Feature Engineering to ręczne projektowanie tego jak wygląda nasze wejście (**X**). Przykładem może być zamiana daty w formacie:

- “2018/12/24”

na 3 osobne feature’y:

- Rok = 2018
- Miesiąc = 12
- Dzień = 24

Można również dodać wartość:

- Dzień wolny od pracy = “Tak”

Teoretycznie przy odpowiedniej ilości danych model sam byłby w stanie nauczyć się tego, że 24 grudnia to dzień wolny od pracy. Jednak podanie mu tej informacji wprost pomaga.

Lista technik inżynierii cech

- Imputacja (uzupełnienie braku danych)
- Obsługa wartości odstających
- Kodowanie One-Hot
- Skalowanie
- Standaryzacja
- Normalizacja

3. Podział danych

Dane walidacyjne

Pracując nad modelem możemy zastanawiać się który z 3 znanych nam algorytmów użyć. Chcielibyśmy mieć możliwość oceny jak nasze zmiany wpływają na jego jakość. Potrzebujemy więc zestawu danych które posłużą nam do jego oceny. Zestaw taki nazywamy danymi walidacyjnymi. Dane te nie powinny zawierać się w danych treningowych. Dzięki temu jeżeli nasz model będzie cierpiał na nadmierne dopasowanie, będziemy w stanie to wykryć na danych walidacyjnych.

Dane testowe

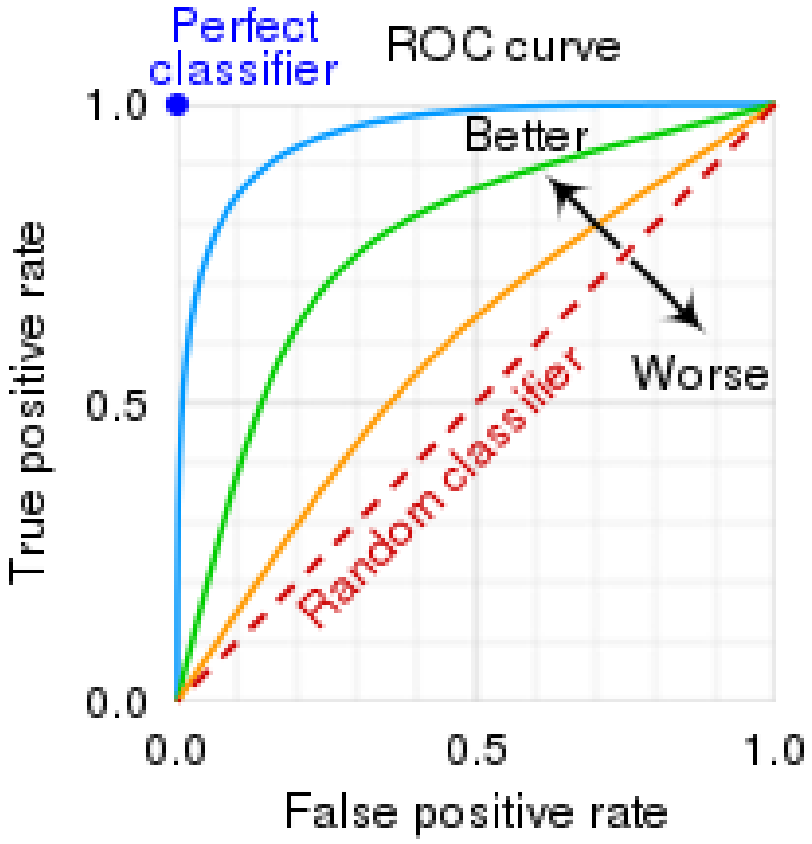
Chcąc odpowiedzieć na pytanie jaka jest faktyczna jakość naszego modelu potrzebujemy zestawu testowego. Część danych nie używamy ani do treningu modeli ani do wyboru najlepszego z nich. Służą nam jedynie na sam koniec do określenia jak dobry jest nasz model.

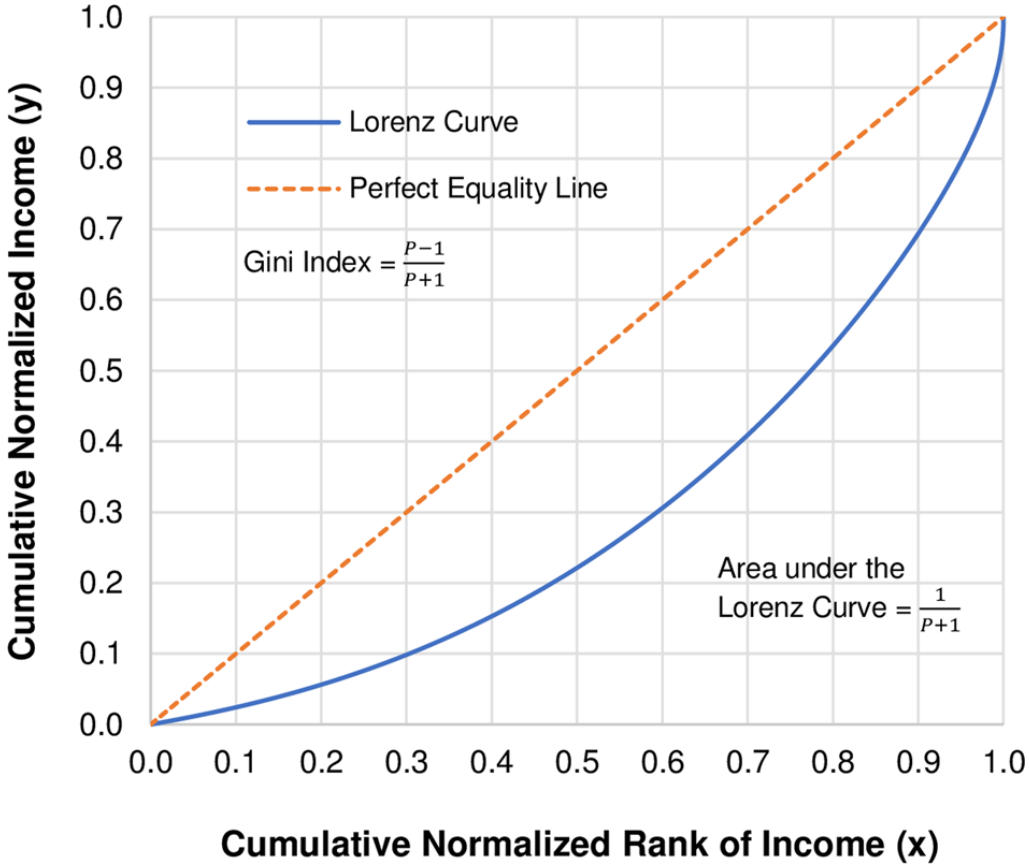
Typowe podziały danych(train, validation, test): 60%/20%/20% albo 80%/10%/10%

4. Miary wydajności modelu

Miary wydajności modelu

Miara wydajności modelu jest liczbą określającą jego jakość. Najbardziej intuicyjnym przykładem jest “dokładność” dla modeli klasyfikujących. Dokładność (ang. accuracy) to stosunek poprawnych do wszystkich odpowiedzi modelu.


Perfect classifier
ROC curve
Better
Worse
Random classifier
True positive rate
False positive rate
1.0
0.5
0.0
0.0
0.5
1.0


Lorenz Curve
Perfect Equality Line
Gini Index = $\frac{P-1}{P+1}$
Area under the Lorenz Curve = $\frac{1}{P+1}$
Cumulative Normalized Income (y)
Cumulative Normalized Rank of Income (x)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

Miary wydajności modelu

Dla regresji:

- Mean Absolute Error (MAE)
- Root-Mean-Square Error (RMSE)

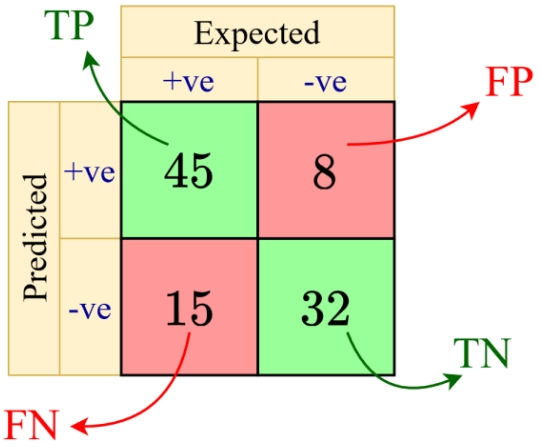
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \hat{X}_i)^2}{n}},$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|X_i - \hat{X}_i\right|$$

Dla klasyfikacji:

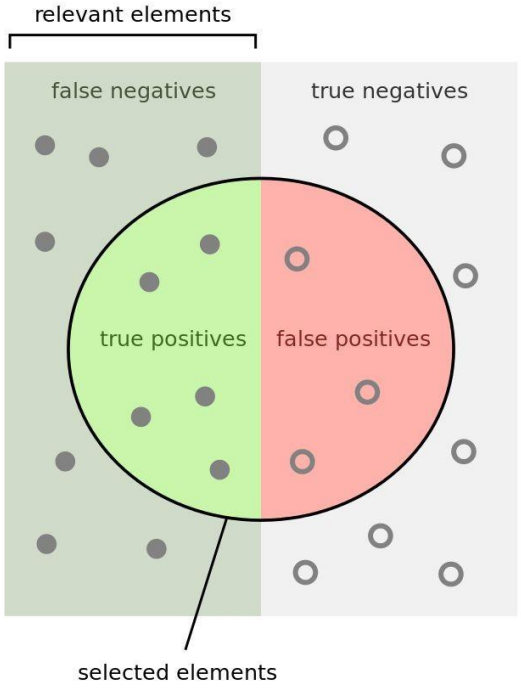
- Dokładność (ang. accuracy)
- Precision
- Recall
- F1

$$F_1 = \left(\frac{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}{2}\right)^{-1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$


TP
Expected
+ve
-ve
FP
Predicted
+ve
45
8
-ve
15
32
TN
FN

Precision i recall

Tworząc model do przewidywania czy ktoś może być chory, zależy nam na wysokim “recall”. W przypadku modelu który typuje firmy do drobiazgowej kontroli skarbowej ważniejszy może być “precision”.



How many selected items are relevant?

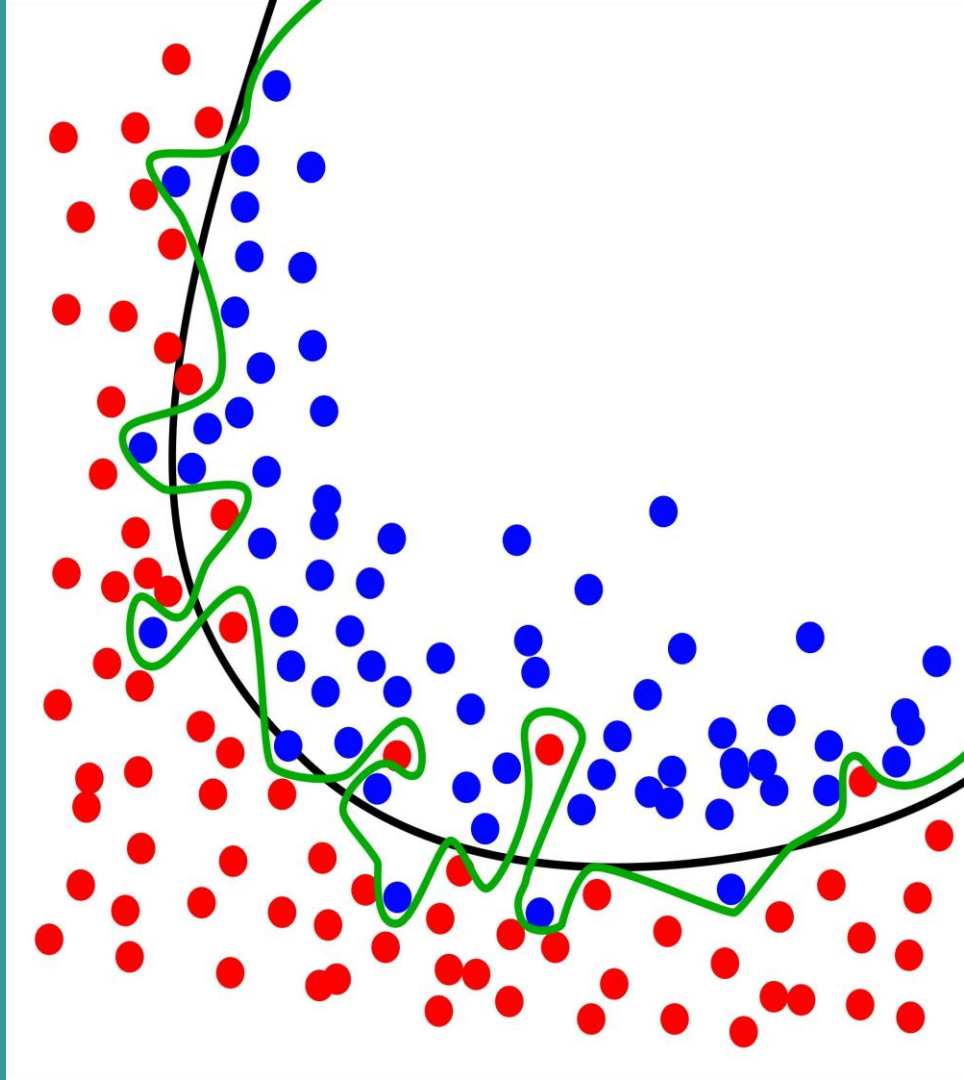
Precision =

How many relevant items are selected?

Recall =

5. Nadmierne dopasowanie

Czarna czy zielona? Która z nich lepiej oddziela te dwie klasy?

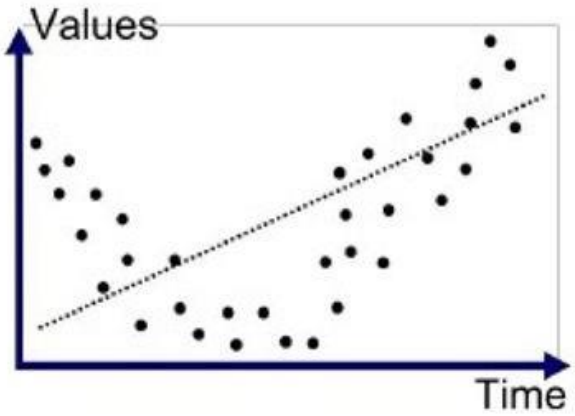


Nadmierne dopasowanie modelu

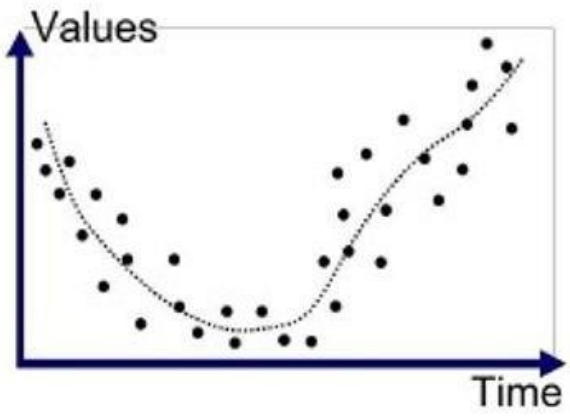
Nadmierne dopasowanie modelu (ang. overfitting) to sytuacja w której nasz model świetnie sobie radzi na danych treningowych ale wypada gorzej na tych których podczas treningu nie widział. Mówimy wtedy, że słabo generalizuje.

Niedopasowanie (ang. underfitting) to sytuacja w której model słabo dopasował się do danych treningowych. I przez to jego skuteczność na danych jest niska.

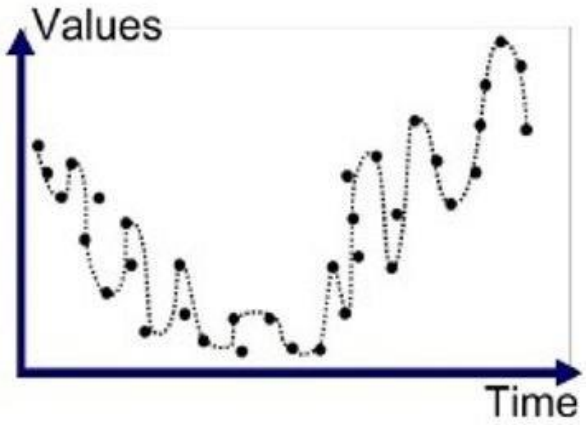
Niedopasowanie i nadmierne dopasowanie



Underfitted



Good Fit/Robust



Overfitted

6. Feature Engineering i tuning parametrów modeli.

Tuning parametrów modeli

To zmienianie właściwości naszego modelu. Dokładnie omówimy to na kolejnych zajęciach poświęconych konkretnym algorytmom. Przykładowe parametry modeli to:

- Głębokość drzewa decyzyjnego
- Ilość drzew decyzyjnych w modelu “lasu”
- Głębokość poszczególnych warstw sieci neuronowej
- Ilość warstw sieci neuronowej

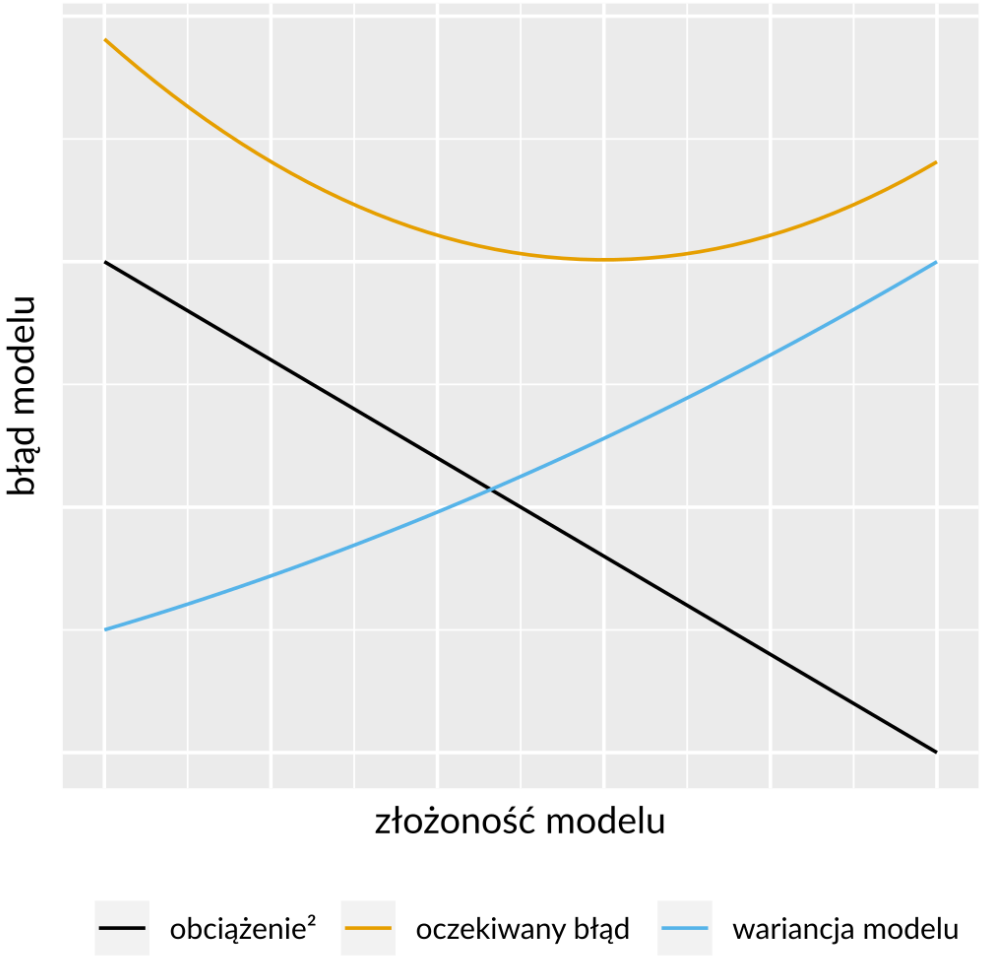
7. Bias-variance tradeoff

Variance

Wariancja jest to wielkość, która mówi nam o tym jak dobrze nasz model szacuje zmienną celu w przypadku użycia innego zbioru testowego. Innymi słowy, wielkość ta opisuje jak dobrze nasz model uchwycił ukryte wzorce w danych i w jakim stopniu jest w stanie przenosić je na inne zbiory danych (nazywamy to generalizacją). Duża wariancja oznacza, że model słabiej radzi sobie w nieznanym środowisku. Najlepiej jeżeli dokładność modelu nie różni się znacząco pomiędzy zbiorem treningowym, a różnymi zbiorami testowymi.

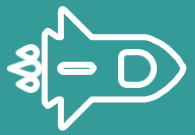
Bias

Modele uczenia maszynowego generalnie działają na pewnych założeniach. Im założenia naszego modelu są prostsze, tym obciążenie naszego modelu jest większe. Im założenie są bardziej skomplikowane, a model bardziej dopasowany do danych treningowych, tym obciążenie jest niższe.


błąd modelu
złożoność modelu
obciążenie²
oczekiwany błąd
wariancja modelu

8.Model workflow

Przykład!



Feature Engineering + bardzo prosty model.

Bez feature engineeringu ale za to bardziej skomplikowany model.

Dzieki