

# Bootcamp Data Science

## Zajęcia 3

Przemysław Spurek

# Regresja

czyli znamy przykładowe wartości  $(x_i, y_i)$ ,  
ktoś nam podaje nowy punkt  $x_0$   
i  
chcemy przewidzieć wartość  $y_0$ .

scikit-learn jest prawdopodobnie najbardziej zaawansowanym pakietem przeznaczonym do nauczania maszynowego, który jest open source (<http://scikit-learn.org>). Zapewnia proste i wydajne narzędzia do wyszukiwania i analizy danych, obejmujące metody zarówno nadzorowane, jak i nienadzorowane.

## Zadanie

Wykonajmy zadanie z poprzedniego zestawu w scikit-learn.

```
https://github.com/przem85/bootcamp/blob/master/statistics/  
D11\_Z01.ipynb
```

## Zadanie

Wykonajmy zadanie z poprzedniego zestawu (dane "Region Alcohol Tobacco") w `scikit-learn`.

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z02.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z02.ipynb)

# bias-variance tradeoff (bias-variance dilemma)

Dla nauczonego modelu istnieje prawdziwa średnia wartość funkcji kosztu  $L$ . Chcemy, aby estymator zwrócił właśnie tę wartość dla dowolnych danych  $D$ . Niestety estymacja będzie różniła się od wartości prawdziwej, a błąd ma trzy składniki:

## bias-variance tradeoff (bias-variance dilemma)

Dla nauczonego modelu istnieje prawdziwa średnia wartość funkcji kosztu  $L$ . Chcemy, aby estymator zwrócił właśnie tę wartość dla dowolnych danych  $D$ . Niestety estymacja będzie różniła się od wartości prawdziwej, a błąd ma trzy składniki:

- **bias** - jeśli dla różnych  $D$  estymator średnio estymuje nieprawdziwą wartość całkowitego kosztu, to odległość tej średniej od prawdziwego całkowitego kosztu nazywana jest biasem,

# bias-variance tradeoff (bias-variance dilemma)

Dla nauczonego modelu istnieje prawdziwa średnia wartość funkcji kosztu  $L$ . Chcemy, aby estymator zwrócił właśnie tę wartość dla dowolnych danych  $D$ . Niestety estymacja będzie różniła się od wartości prawdziwej, a błąd ma trzy składniki:

- **bias** - jeśli dla różnych  $D$  estymator średnio estymuje nieprawdziwą wartość całkowitego kosztu, to odległość tej średniej od prawdziwego całkowitego kosztu nazywana jest biasem,
- **variance** - jeśli dla różnych  $D$  estymator zwraca bardzo podobne liczby, to variance jest mała, jeśli natomiast estymata bardzo zależy od konkretnego  $D$ , to variance jest duża,



# bias-variance tradeoff (bias-variance dilemma)

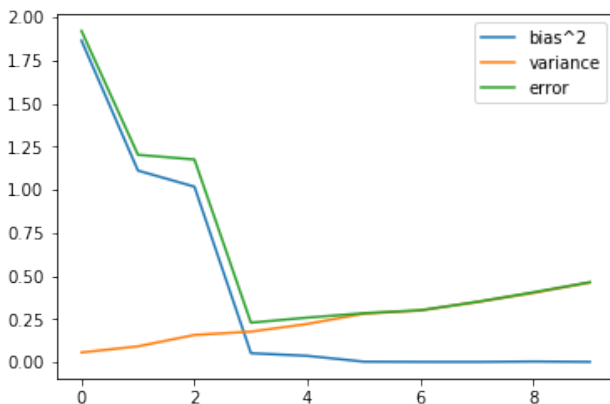
Dla nauczonego modelu istnieje prawdziwa średnia wartość funkcji kosztu  $L$ . Chcemy, aby estymator zwrócił właśnie tę wartość dla dowolnych danych  $D$ . Niestety estymacja będzie różniła się od wartości prawdziwej, a błąd ma trzy składniki:

- **bias** - jeśli dla różnych  $D$  estymator średnio estymuje nieprawdziwą wartość całkowitego kosztu, to odległość tej średniej od prawdziwego całkowitego kosztu nazywana jest biasem,
- **variance** - jeśli dla różnych  $D$  estymator zwraca bardzo podobne liczby, to variance jest mała, jeśli natomiast estymata bardzo zależy od konkretnego  $D$ , to variance jest duża,
- **noise** - szum zawarty w liczbach oznaczonych  $y_i^{true}$ ; w  $y_i^{pred}$  oczywiście nie ma szumu, bo przypominam, że nauczony model jest deterministyczny.

Cała sztuka polega na znalezieniu złotego środka, gdzie suma bias + variance jest najmniejsza, a więc nasza predykcja najdokładniejsza.

# bias-variance tradeoff (bias-variance dilemma)

<https://github.com/przem85/bootcamp/blob/master/statistics/BV.ipynb>



# Underfitting vs. Overfitting

Mówiliśmy o błędzie modelu oraz estymatora całkowitego kosztu.

Bias-variance dilemma dotyczy dowolnej estymacji!

Przykłady:

- $mean(x_1, \dots, x_n) = \bar{x} = \frac{1}{n} \sum x_i$  - estymacja średniej z próby,
- $s^2(x_1, \dots, x_n) = \frac{1}{n-1} \sum (x_i - \bar{x})^2$  - estymacja wariancji z próby.

Wtedy także zachodzi ( $\hat{\theta}$  - oznaczenie estymatora):

$$MSE(\hat{\theta}(X)) = var(\hat{\theta}) + (B(\hat{\theta}))^2$$

# Underfitting vs. Overfitting

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z03.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z03.ipynb)

Uczenie parametrów funkcji predykcyjnej i testowanie jej na tych samych danych jest **błędem metodologicznym**:

*model, który po prostu powtarzałby etykiety próbek, które właśnie widział, miałby doskonały wynik, ale nie przewidywałby niczego przydatnego na danych, których jeszcze nie widział.*

# Underfitting vs. Overfitting

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z03.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z03.ipynb)

Uczenie parametrów funkcji predykcyjnej i testowanie jej na tych samych danych jest **błędem metodologicznym**:

*model, który po prostu powtarzałby etykiety próbek, które właśnie widział, miałby doskonały wynik, ale nie przewidziałby niczego przydatnego na danych, których jeszcze nie widział.*

Sytuacja ta nazywana jest **Overfitting**. Aby tego uniknąć, podczas wykonywania eksperymentu (nadzorowanego) powinniśmy podzielić dane na dwie grupy  $X_{train}$ ,  $X_{test}$ .

# Underfitting vs. Overfitting

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z03.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z03.ipynb)

Uczenie parametrów funkcji predykcyjnej i testowanie jej na tych samych danych jest **błędem metodologicznym**:

*model, który po prostu powtarzałby etykiety próbek, które właśnie widział, miałby doskonały wynik, ale nie przewidziałby niczego przydatnego na danych, których jeszcze nie widział.*

Sytuacja ta nazywana jest **Overfitting**. Aby tego uniknąć, podczas wykonywania eksperymentu (nadzorowanego) powinniśmy podzielić dane na dwie grupy  $X_{train}$ ,  $X_{test}$ .

Zauważ, że słowo “eksperyment” nie oznacza wyłącznie wykorzystania w celach akademickich, ponieważ nawet w komercyjnych warunkach uczenie maszyn rozpoczyna się eksperymentalnie.

# Zbiór testowy i zbiór treningowy

[https://github.com/przem85/bootcamp/blob/master/statistics/plot\\_underfitting\\_overfitting.ipynb.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/plot_underfitting_overfitting.ipynb.ipynb)

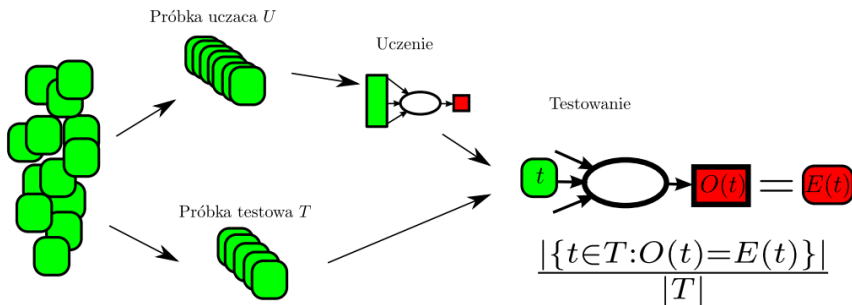
[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z04.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z04.ipynb)

Dane uczące są losowo dzielone na dwa rozłączne zbiory:

- próbkę uczącą  $X_{train}$ ,
- próbkę testową  $X_{test}$ .

Model jest uczony za pomocą próbki uczącej, natomiast jest oceniany na podstawie próbki testowej.

# Zbiór testowy i zbiór treningowy





Uwagi i niebezpieczeństwa:

- czasami na wynik większy wpływ ma stosunek  $\frac{|X_{train}|}{|X_{train} \cup X_{test}|}$ , niż zaimplementowany algorytm,
- rozsądnym minimum dla wielkości  $X_{train}$  jest około  $\frac{1}{4}$  całego zbioru,
- z drugiej strony  $X_{train}$  nie powinno być większe niż  $\frac{9}{10}$  całego zbioru,
- podając wynik walidacji zawsze należy podać proporcje w jakich podzielono zbiór.

# k-fold cross-validation

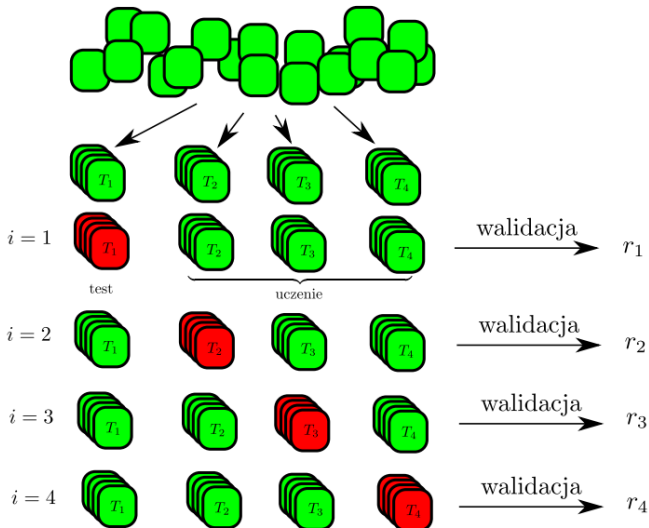
[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z05.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z05.ipynb)

- dane uczące są losowo dzielone na  $k$  rozłącznych zbiorów:  $T_1, \dots, T_k$ ,
- zbiory powinny być równoliczne (lub różnić się o maksymalnie 1 element, jeżeli nie da się podzielić dokładnie),
- dla  $i = 1 \dots k$  powtarzamy
  - uczymy sieć na zbiorze uczącym

$$T_1 \cup \dots \cup T_{i-1} \cup T_{i+1} \cup T_{i-1} \cup \dots T_k$$

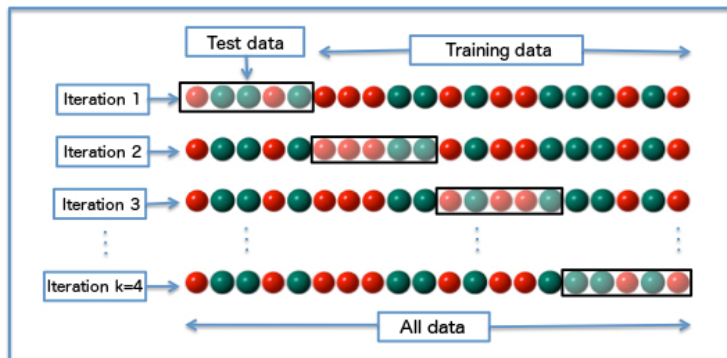
- testujemy tak nauczony model na danych  $T_i$  (na tych danych model nie był uczony),
  - zapamiętujemy rezultat jako  $r_i$
- zależnie od ilości miejsca podajemy wszystkie rezultaty  $r_i$  lub ich średnią medianę, minimum, maximum.

# k-fold cross-validation



# k-fold cross-validation

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z06.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z06.ipynb)



# Leave One Out

Leave One Out to odmiana walidacji krzyżowej, w której  $k =$  ilość elementów w  $T$

- dla  $i = 1, \dots, k$  powtarzamy
  - uczymy sieć na zbiorze uczącym

$$T \setminus T_i$$

- testujemy tak nauczony model na pozostałym przykładzie  $T_i$ ,
  - zapamiętujemy rezultat jako  $r_i$
- podajemy średnią medianę, minimum, maximum z  $r_i$ .
- można stosować w przypadku małej ilości danych w zbiorze  $T$

## Regularyzacja

Przez regularyzację rozumiemy faworyzowanie “prostszych” hipotez w ramach danego zbioru hipotez (modelu). Mówiąc inaczej, regularyzacja zwiększa bias kosztem wariancji, a więc jej stosowanie ma sens tylko wtedy, gdy przypuszczamy, że to wariancja jest dominującym składnikiem błędu całkowitego.

Regularyzacja jest zgodna z wymaganiem, by zawsze wybierać **najprostszy model**, który wystarczająco **dobrze tłumaczy zjawiska**:

- **Brzytwa Ockhama** – Nie mnoż bytów ponad konieczność (przypisywane Williamowi of Ockham, c. 1287-1347)
- **Paul Dirac (1902-1984)** – Teoria, która jest piękna matematycznie, ma większą szansę by być prawdziwa od tej brzydkiej, która dobrze pasuje do danych

Celem regresji liniowej jest zminimalizowanie sumy kwadratów odległości:

$$\min_w \|Aw - y\|^2.$$

Wynikiem regresji liniowej jest model:

$$\hat{y} = w_0 + w_1 x_i + \dots + w_p x_p.$$

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z07.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z07.ipynb)

Ridge regression próbuje rozwiązać niektóre problemy klasycznej regresji liniowej, nakładając kary za wykorzystanie większej ilości współrzędnych. Ridge regression minimalizuje:

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2,$$

gdzie:

$$||w||_2 = \sqrt{w_1^2 + \dots + w_n^2}.$$

Parametr  $\alpha \geq 0$  jest parametrem złożoności, który kontroluje wielkość kary; im większa jest wartość  $\alpha$ , tym większa kara.



[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z09.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z09.ipynb)

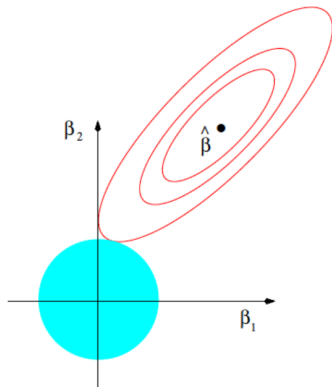
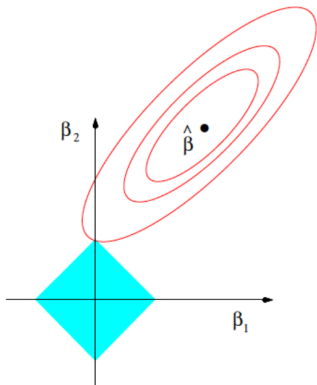
Lasso regression próbuje rozwiązać niektóre problemy klasycznej regresji liniowej, nakładając kary za wykorzystanie większej ilości współrzędnych. Lasso regression minimalizuje:

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_1^2,$$

gdzie:

$$||w||_1 = \sum_{i=1}^n |w_i|.$$

# Lasso Regression



[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z11.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z11.ipynb)

Elastic Net próbuje rozwiązać niektóre problemy klasycznej regresji liniowej, nakładając kary za wykorzystanie większej ilości współrzędnych. Elastic Net minimalizuje:

$$\min_w \frac{1}{2n_{\text{samples}}} \|Xw - y\|_{Fro}^2 + \alpha \|w\|_{21},$$

gdzie  $Fro$  oznacza Frobenius norm:

$$\|A\|_{Fro} = \sqrt{\sum_{ij} a_{ij}^2}$$

oraz  $\ell_1\ell_2$ :

$$\|A\|_{21} = \sum_i \sqrt{\sum_j a_{ij}^2}$$

Parametr  $\alpha$  można dobrać za pomocą procedury cross-validation.

## Przykład

Praca z danymi wielowymiarowymi niesie ze sobą wiele niebezpieczeństw. Rozważmy następujący przykład: w golfa przeważnie grają bogaci ludzie i wiadomo, że przeciętnie liczba dzieci spada wraz ze wzrastającym dochodem.

Innymi słowy, mamy dość silną ujemną korelację pomiędzy grą w golfa, a liczbą dzieci, i można by się skusić (fałszywie) wyciągnąć wniosek, że gra w golfa zmniejsza płodność. Ale w rzeczywistości są to wyższe dochody, które powodują oba skutki.

*Kaplan, D. (2009). Statistical modeling: A fresh approach. St Paul: Macalester College.*

# Regresja wielowymiarowa

Regresja wieloliniowa (lub regresja wielokrotna) jest prostym przedłużeniem prostej regresji liniowej.

Założmy, że dla dwóch zmiennych  $w_i$  i  $x_i$  chcemy nauczyć się przewidywać trzecią ( $y_i$ ). W takiej sytuacji model wygląda tak:

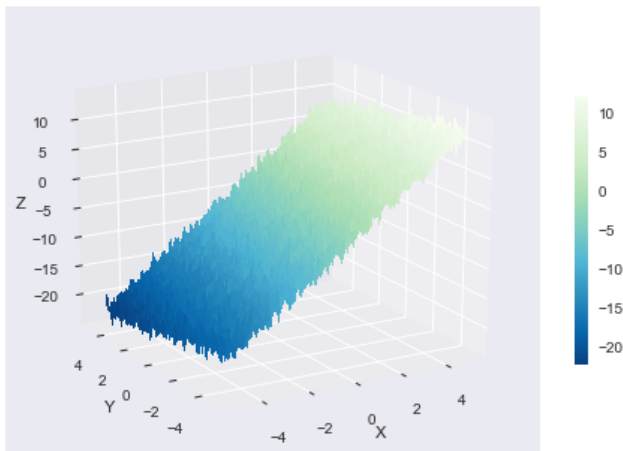
$$y_i = \beta_0 + \beta_1 w_i + \beta_2 x_i + \epsilon_i.$$

Dla 7 punktów mamy:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & w_1 \\ 1 & x_2 & w_2 \\ 1 & x_3 & w_3 \\ 1 & x_4 & w_4 \\ 1 & x_5 & w_5 \\ 1 & x_6 & w_6 \\ 1 & x_7 & w_7 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \end{bmatrix},$$

# Regresja wielowymiarowa

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z12.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z12.ipynb)



[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z18.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z18.ipynb)

[https://github.com/przem85/bootcamp/blob/master/statistics/D12\\_Z06.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D12_Z06.ipynb)

[https://github.com/przem85/bootcamp/blob/master/statistics/D11\\_Z19.ipynb](https://github.com/przem85/bootcamp/blob/master/statistics/D11_Z19.ipynb)