Klasyfikatory

Zgromadzone dane

- Tabela
  - Wiersze : Kolejne obserwacje
  - Kolumny : zmienna prognozowana (objaśniana) (Y)
    zmienne objaśniające (**X**)
- Typy danych
  - Nominalne (binarne)
  - Porządkowe
  - Ciagłe
  - Braki danych

Braki danych

- Braki danych są jednym z najważniejszych problemów (i najczęstszych) na jaki możemy natrafić budując model.
- Aby móc im poprawnie przeciwdziałać musimy odpowiednio zbadać i zrozumieć strukturę zebranych danych.
- Mechanizmy
  - Mechanizm całkowicie losowy (**MCAR**: *Missing completely at random*)
  - Mechanizm losowy (**MAR**: *Missing at random*)
  - Mechanizm nielosowy (**MNAR** *Missing not at random*)
- Uwaga
  - Wybór procedury radzenia sobie z brakami danych zależy od typu modelu na którym pracujemy (np. szeregi czasowe).

Braki danych

- **Jak sobie radzić?**
- Usuwanie danych
  - **Usuwanie wszystkich jednostek obserwacji z analiz**
  - **Usuwanie jednostek obserwacji z analiz parami**
  - **Usuwanie zmiennych**
- Zastępowanie braków danych
  - Za pomocą statystyk (średniej/mediany/dominanty)
  - Za pomocą modelu
    - **Imputacja nieparametryczna**
    - **Imputacja regresyjna**
    - **Metoda największej wiarygodności**
    - **Wielokrotne imputacje**
- Traktowanie braku danych jako dodatkowej informacji, którą niesie ze sobą model
  - Traktujemy braki danych jako zmienne binarne – szczególnie wygodne w przypadku zmiennych jakościowych

Zagadnienie klasyfikacji

- Binarna zmienna objaśniana
- Szukamy dowolnej funkcji zmiennych objaśniających $f(x)$ takiej, że:

$$f(\mathbf{X}_1) > f(\mathbf{X}_2) \Leftrightarrow \Pr(Y_1 = 1) > \Pr(Y_2 = 1)$$

- Klasyczne modele w których zmienne objaśniane są wyrażalne liczbowo i nie posiadają braków danych
  - Regresja liniowa
  - Regresja logistyczna

Liniowy model prawdopodobieństwa

- Standardowa postać funkcyjna:

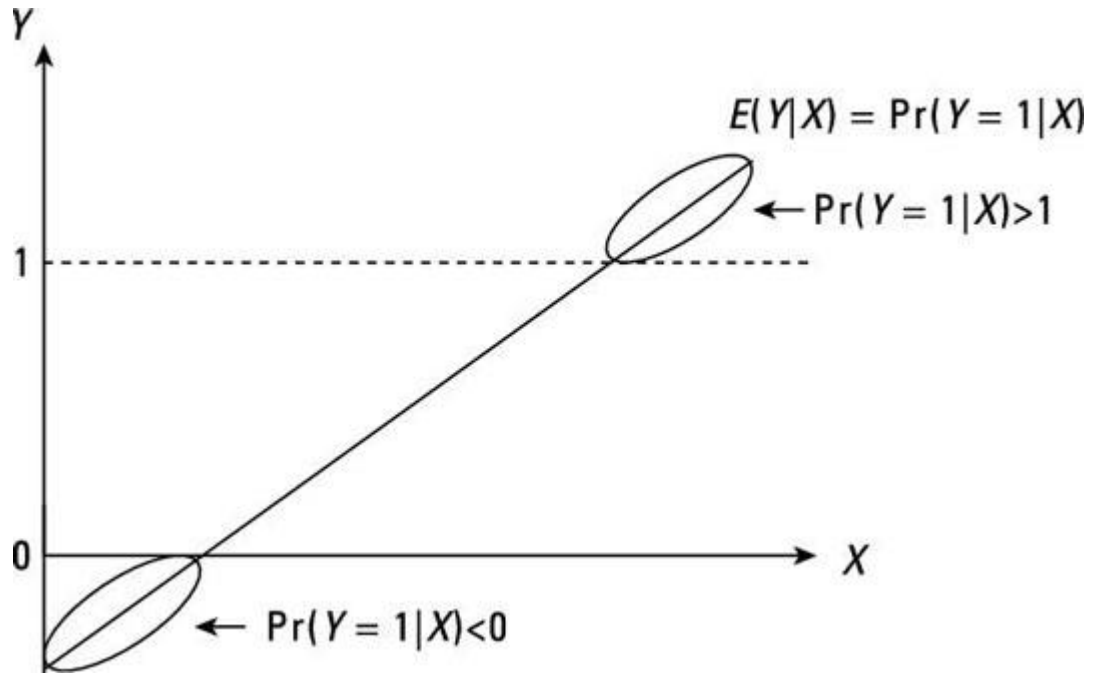
$$f(X) = \mathrm{P}(Y = 1|X)$$
$$= \mathrm{B}_0 + \mathrm{B}_1 x_1 + \cdots + \mathrm{B}_n x_n$$

- Uwagi
  - Funkcja może być dowolna (niekoniecznie liniowa)
  - Nie będziemy zajmowali się własnościami statystycznymi
- Sposób wyznaczania parametrów na podstawie n-elementowego zbioru uczącego:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left\{ \sum_{i=1}^{n} \left(y_i - f(\mathbf{X}_i) \right)^2 \right\}$$

Liniowy model prawdopodobieństwa

$$\mathrm{P}(Y = 1|X) = \mathrm{B}_0 + \mathrm{B}_1 x_1 + \cdots + \mathrm{B}_n x_n$$



Regresja logistyczna

- Standardowa postać funkcyjna:

$$g(x) = \mathrm{B}_0 + \mathrm{B}_1 x_1 + \cdots + \mathrm{B}_n x_n$$

$$f(x) = \mathrm{P}(Y = 1|X) = \frac{\exp(g(X))}{1 + \exp(g(X))}$$

- Sposób wyznaczania parametrów na podstawie n-elementowego zbioru uczącego:

$$\mathbf{a} = \arg\max_{\mathbf{a}} \left\{ \sum_{i=1}^{n} y_i \ln\left(f(\mathbf{X})\right) + \left(1 - y_i\right) \ln\left(1 - f(\mathbf{X})\right) \right\}$$

Regresja logistyczna

- Własności:
  - Zawsze w przedziale (0,1)
  - Można **interpretować** jako prawdopodobieństwo
- Iloraz szans jest równy funkcji bazowej $g(X)$:

$$\ln\left(\frac{f(X)}{1-f(X)}\right) = g(X)$$

- Uwaga
  - Zamiast dystrybuanty rozkładu logistycznego można użyć innej
  - Dla dystrybuanty standardowego rozkładu normalnego model nazywany jest probitowym

Regresja logistyczna

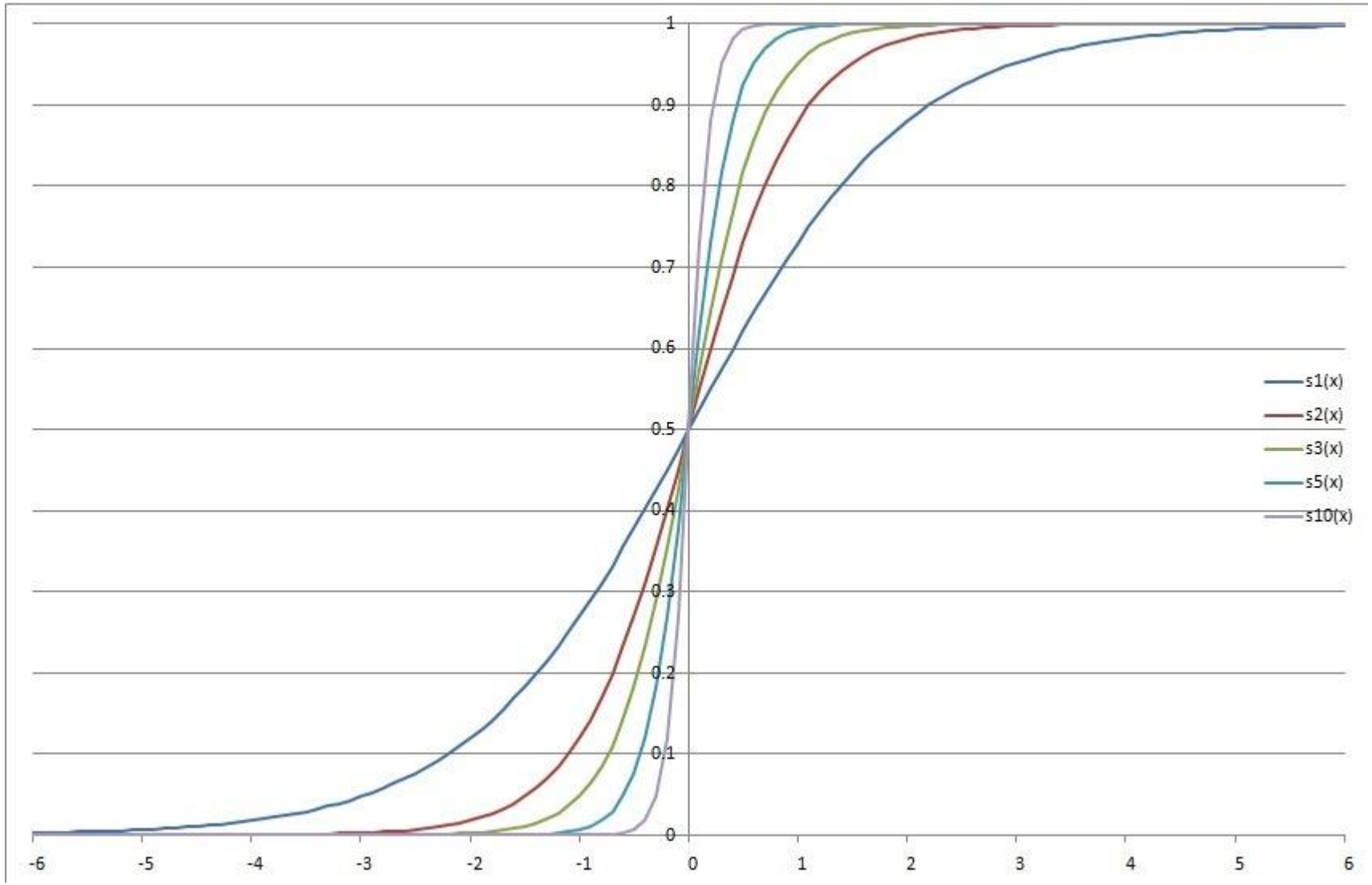
- **Probit**:

$$\mathrm{P}(Y = 1|X) = \Phi(\mathrm{B}_0 + \mathrm{B}_1 x_1 + \cdots + \mathrm{B}_n x_n)$$

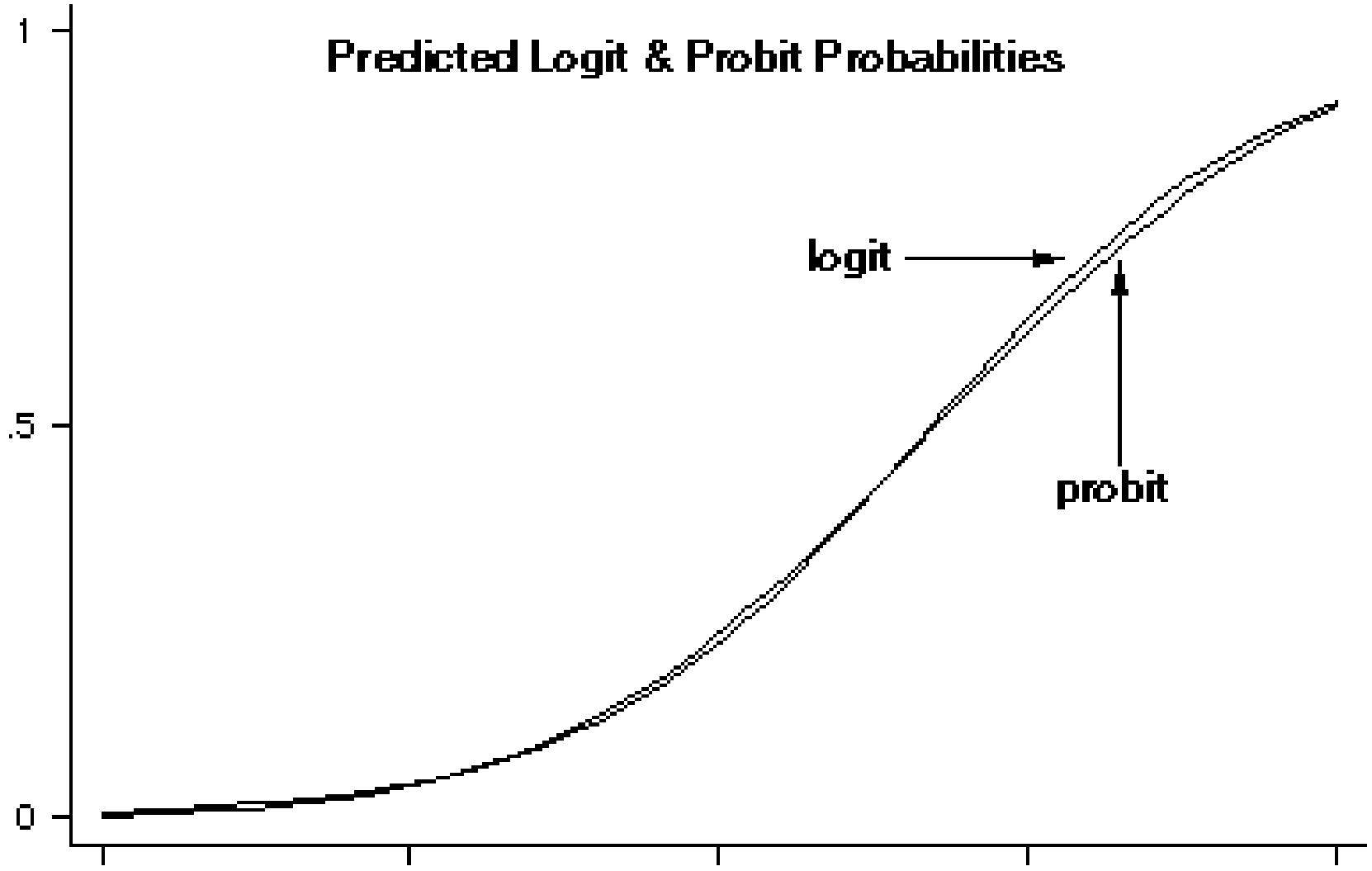
- **Logit**:

$$\mathrm{P}(Y = 1|X) = \frac{e^{\mathrm{B}_0 + \mathrm{B}_1 x_1 + \cdots + \mathrm{B}_n x_n}}{1 + e^{\mathrm{B}_0 + \mathrm{B}_1 x_1 + \cdots + \mathrm{B}_n x_n}}$$

Funkcja sigmoidalna



Funkcja sigmoidalna



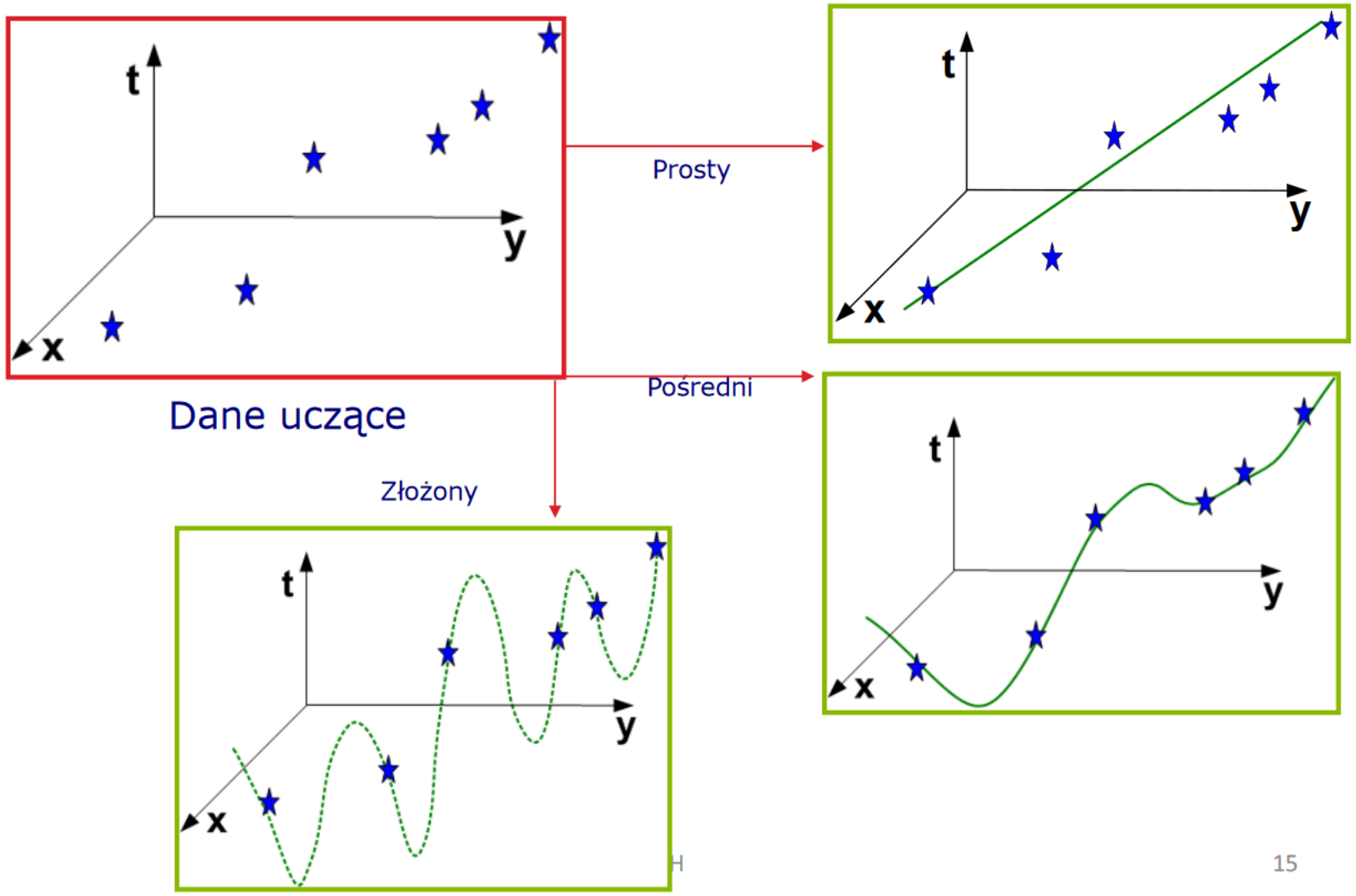
Ocena jakości modelu

- Określenie tego w jaki sposób mierzyć błąd modelu i jaka jest jego najwyższa akceptowana wartość jest (prawie) zawsze koniecznym pierwszym krokiem.
- Przede wszystkim dlatego, że nierozerwalnie wiąże się z koniecznością zrozumienia jaki jest cel budowy danego modelu i przez to wpływa na to w jaki sposób ta budowa będzie przebiegała (jaki algorytm uczący zostanie wykorzystany, jak duży i jak skonstruowany będzie zbiór uczący, etc.).

Ocena jakości modelu

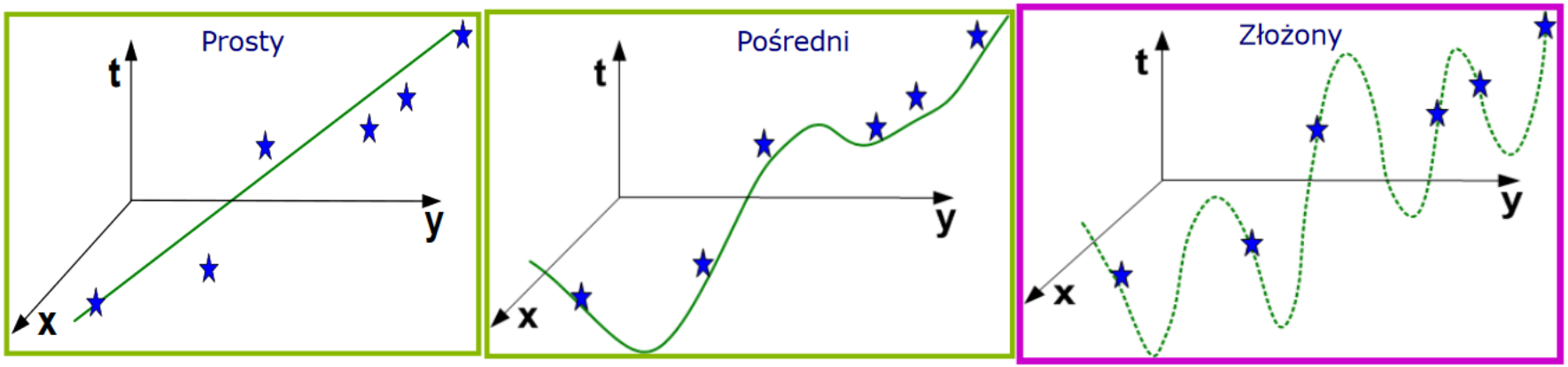
- Jest to szczególnie ważne gdy interesuje nas odpowiedź na pytanie: **Jaki model powinniśmy wybrać?**

Ocena jakości modelu

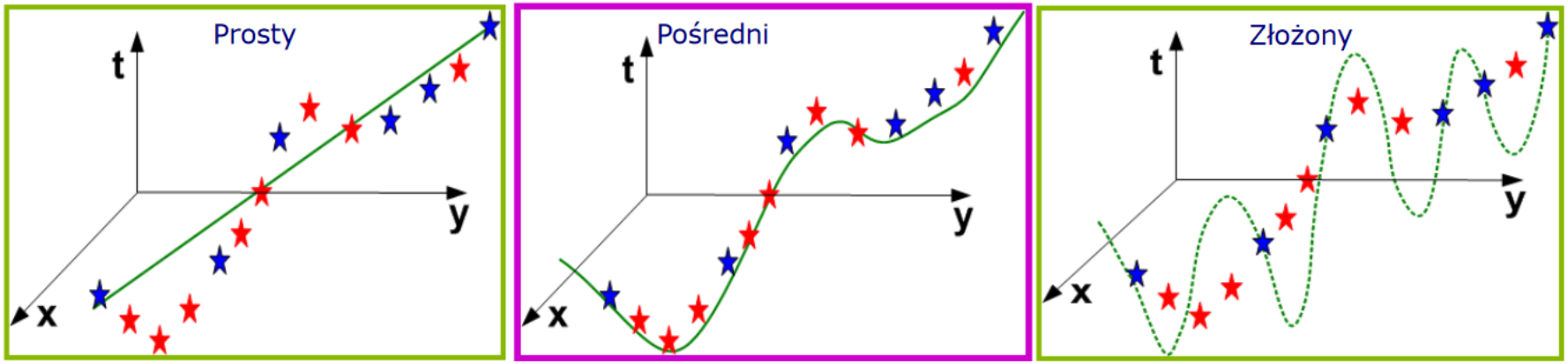


Ocena jakości modelu

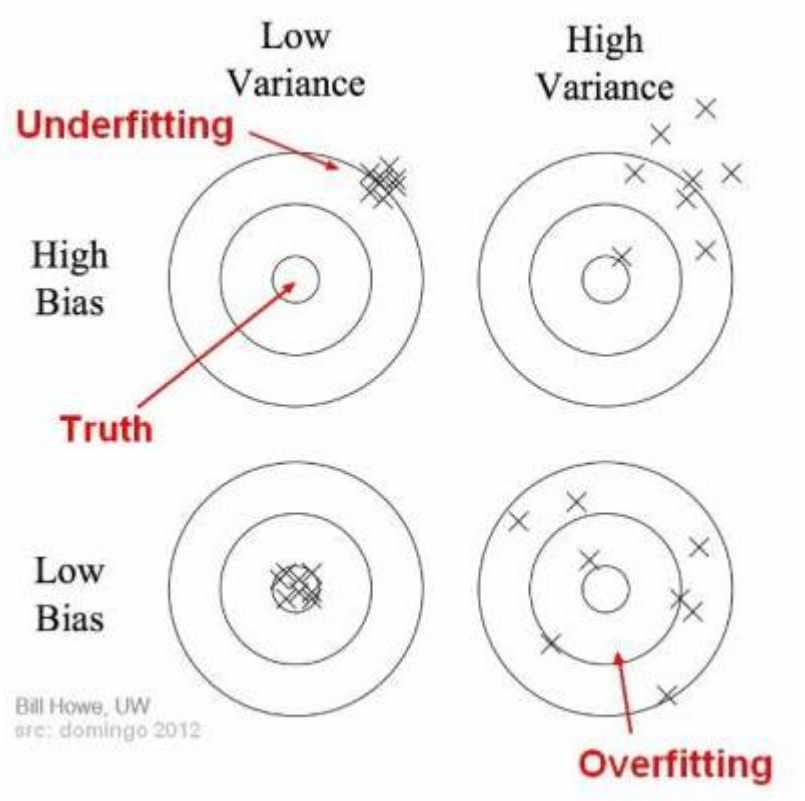
Minimalizacja błędu uczenia

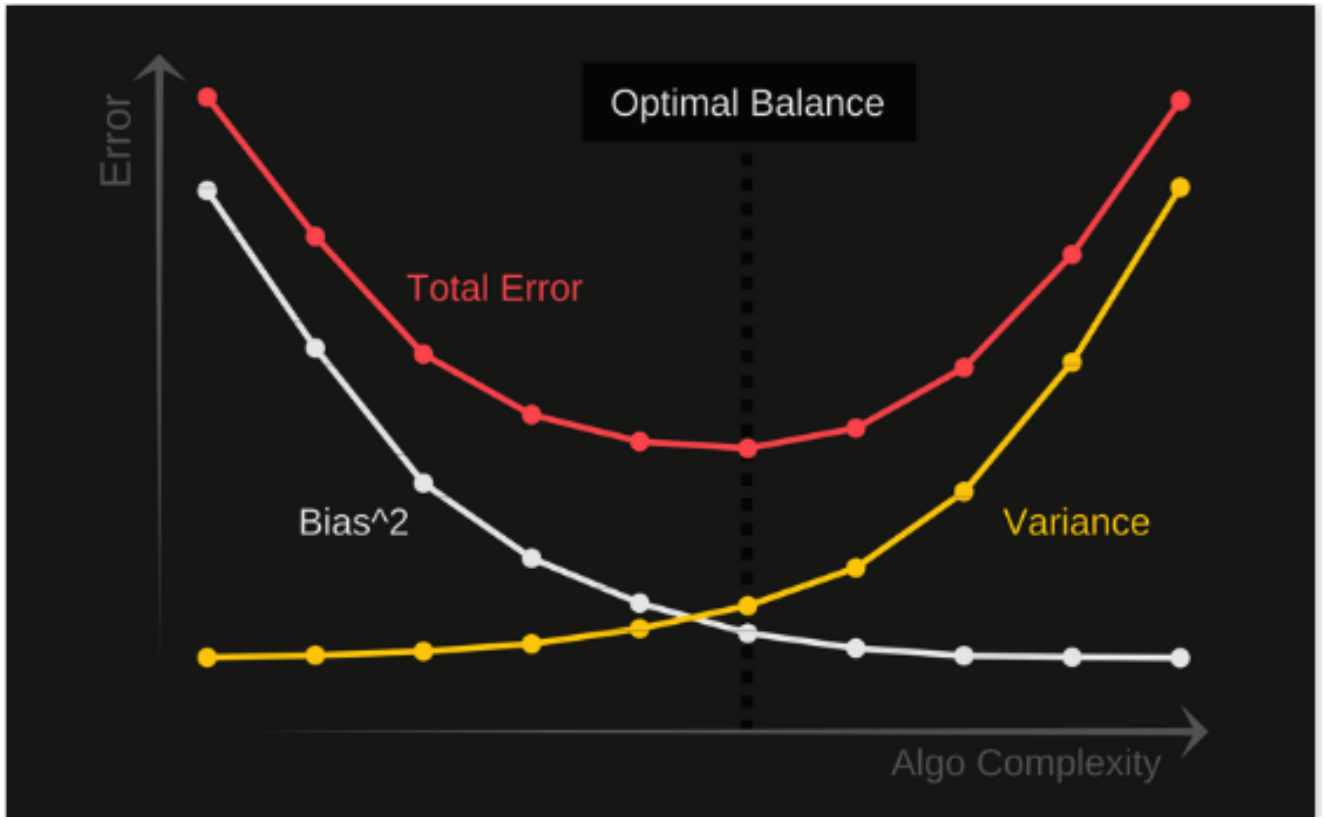


Minimalizacja błędu prognozy



Bias-Variance Tradeoff





Źródło: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

Ocena jakości modelu

- Abyśmy mogli odpowiednio ocenić, wyspecyfikować i wreszcie wybrać odpowiedni model musimy zastanowić się nad dwiema podstawowymi kwestiami:
  - Odpowiednią miarą jakości (**metryką**) modelu
  - Procedurą uczenia, oceny i wyboru modelu

Wybór odpowiedniej metryki

Przykładowe metryki:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\big(G(x) - f(x)\big)^2$$

$$MISE = E\big||G - f|\big|_2^2 = E[\int \big(G(X) - f(x)\big)^2\, dx]$$

Wybór odpowiedniej metryki

Przykładowe metryki:

- Macierz klasyfikacji:

		Klasa prawdziwa: 1	Klasa prawdziwa: 0
Klasa prognozowana	1	**True positive (TP)**	**False positive (FP)**
	0	**False negative (FN)**	**True negative (TN)**

Wybór odpowiedniej metryki

Przykładowe metryki:

- Macierz klasyfikacji:

Klasa prawdziwa

Klasa prognozowana	1	0
[f(X)>T] = 1	**True positive (TP)**	**False positive (FP)**
[f(X)>T] = 0	**False negative (FN)**	**True negative (TN)**

- Gdzie T nazywamy progiem odcięcia

Wybór odpowiedniej metryki

Przykładowe metryki:

- Macierz klasyfikacji:

		Klasa prawdziwa: 1	Klasa prawdziwa: 0
Klasa prognozowana	1	True positive (TP)	False positive (FP)
	0	False negative (FN)	True negative (TN)

Trafność (*Accuracy*): $acc = \frac{TP+TN}{P+N}$

Wybór odpowiedniej metryki

Przykładowe metryki:

- Macierz klasyfikacji:

		Klasa prawdziwa: 1	Klasa prawdziwa: 0
Klasa prognozowana	1	True positive (TP)	False positive (FP)
	0	False negative (FN)	True negative (TN)

Czułość (Sensitivity – Recall – True Positive Rate): $TPR = \frac{TP}{TP+FN}$

Specyficzność (Specifity – True Negative Rate): $TNR = \frac{TN}{TN+FP}$

Precyzja (Precision – Positive Predictive Value): $PPV = \frac{TP}{TP+FP}$

Wybór odpowiedniej metryki

Przykładowe metryki:

- F - score:

$$F = \frac{2PPV * TPR}{PPV + TPR}$$

Czułość (Sensitivity – Recall – True Positive Rate): $TPR = \frac{TP}{TP+FN}$

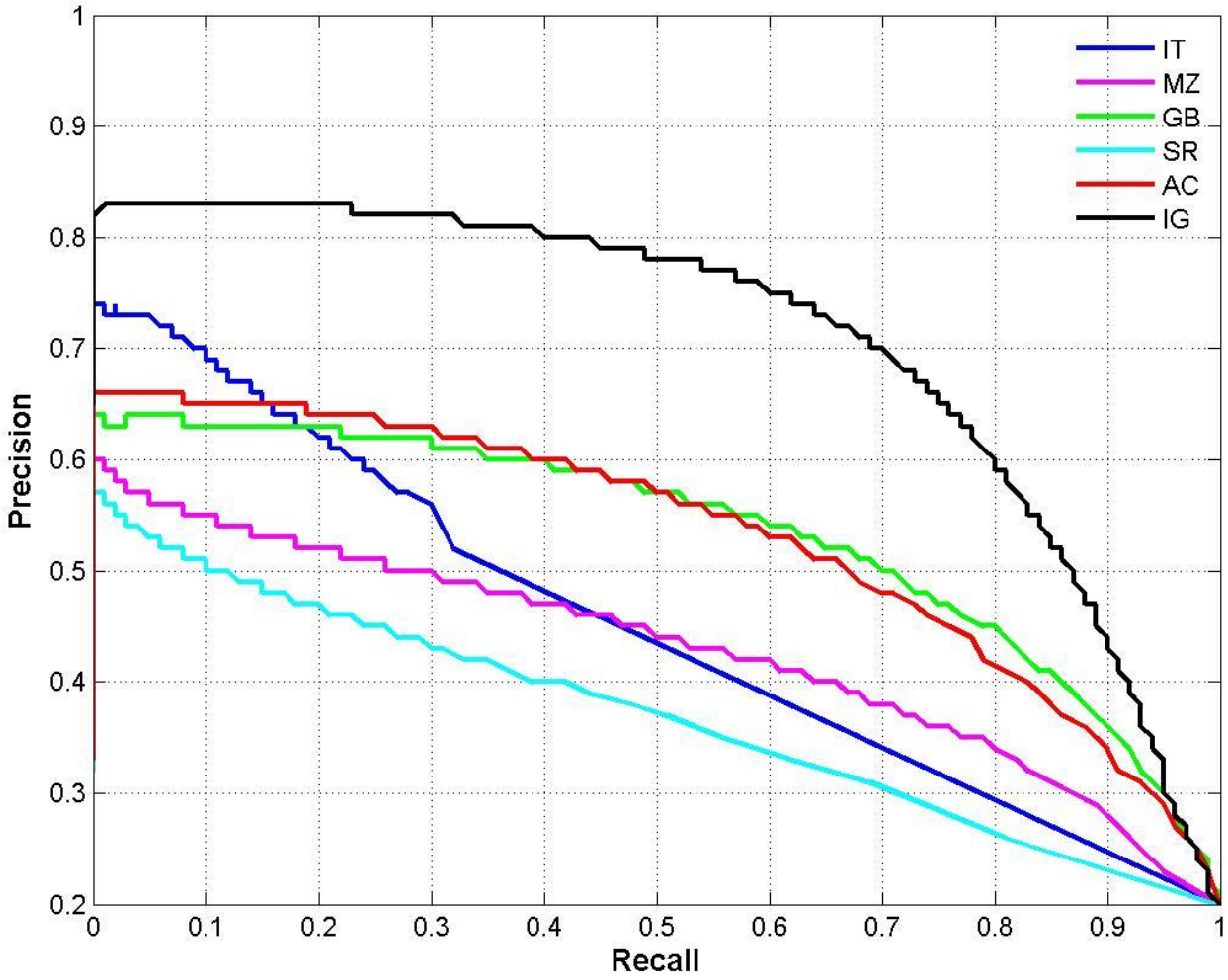
Specyficzność (Specifity – True Negative Rate): $TNR = \frac{TN}{TN+FP}$

Precyzja (Precision – Positive Predictive Value): $PPV = \frac{TP}{TP+FP}$

Wybór odpowiedniej metryki

Przykładowe metryki:

- Krzywa PR:



Czułość (Sensitivity – Recall – True Positive Rate): $TPR = \frac{TP}{TP+FN}$

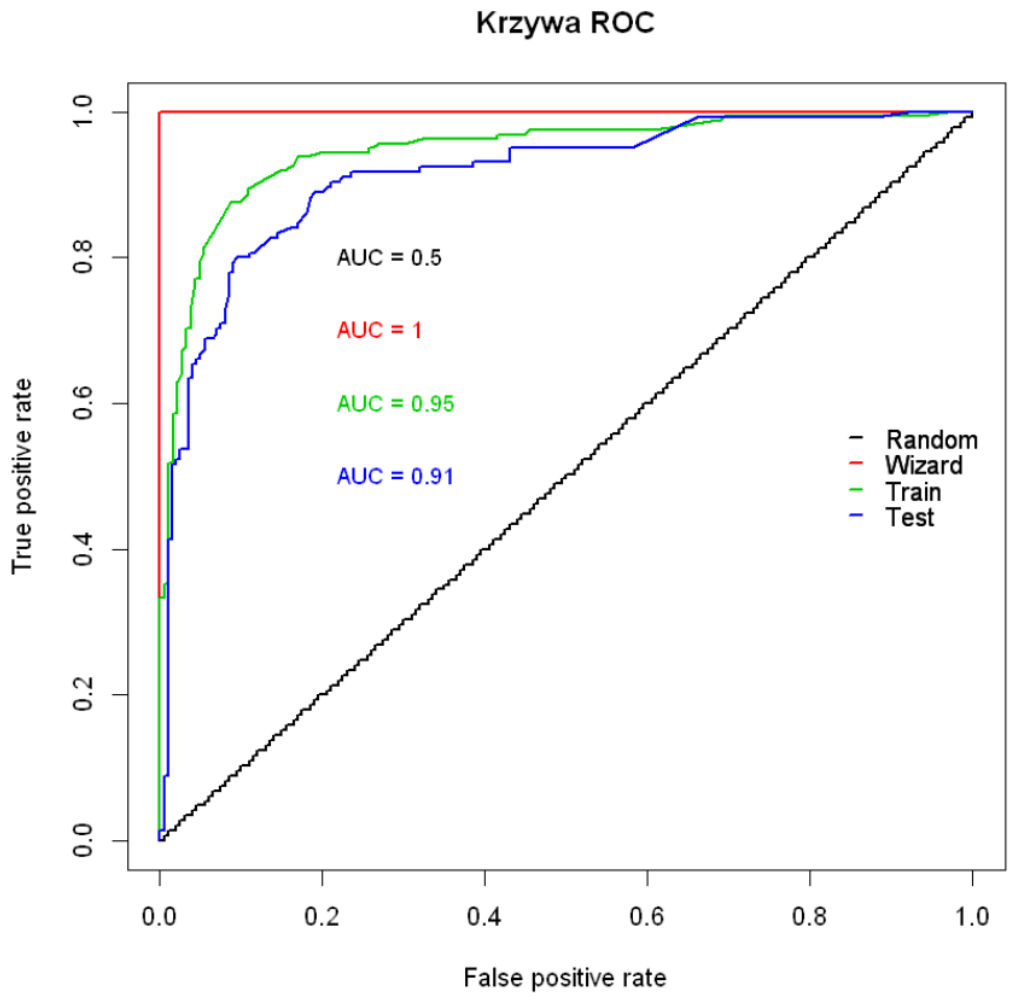
Specyficzność (Specifity – True Negative Rate): $TNR = \frac{TN}{TN+FP}$

Precyzja (Precision – Positive Predictive Value): $PPV = \frac{TP}{TP+FP}$

Wybór odpowiedniej metryki

Przykładowe metryki:

- Krzywa ROC:



Czułość (Sensitivity – Recall – True Positive Rate): $TPR = \frac{TP}{TP+FN}$

Specyficzność (Specifity – True Negative Rate): $TNR = \frac{TN}{TN+FP}$

Precyzja (Precision – Positive Predictive Value): $PPV = \frac{TP}{TP+FP}$

Wybór odpowiedniej metryki

- Wybór metryki zależy bezpośrednio od tego czemu służyć ma tworzony model.
- Inaczej będziemy traktowali błąd w przypadku testów wykrywających występowanie ciężkiej choroby (np. raka) a inaczej w przypadku modelu klasyfikującego zdjęcia psów i kotów.
- Dodatkowo pomiar błędu zależeć może od wielu innych czynników (np. generowanego finansowego zysku/straty).

Wybór odpowiedniej metryki

https://www.theguardian.com/technology/video/2017/sep/12/apple-iphone-x-facial-recognition-face-id-fail-launch-video

- Ale:

Wybór odpowiedniej metryki

https://www.theguardian.com/technology/video/2017/sep/12/apple-iphone-x-facial-recognition-face-id-fail-launch-video

- Ale:

iPhone X racism row: Apple's Face ID fails to distinguish between Chinese users

https://www.mirror.co.uk/tech/apple-accused-racism-after-face-11735152

Optymalizacja progu odcięcia

Przykładowe metryki:

- Macierz klasyfikacji:

		Klasa prawdziwa: 1	Klasa prawdziwa: 0
Klasa prognozowana	[f(X)>T] = 1	**True positive (TP)**	**False positive (FP)**
	[f(X)>T] = 0	**False negative (FN)**	**True negative (TN)**

- Gdzie T nazywamy progiem odcięcia

Optymalizacja progu odcięcia

- Macierz klasyfikacji:

Klasa prawdziwa

Klasa prognozowana	1	0
[f(X)>T] = 1	**True positive (TP)**	**False positive (FP)**
[f(X)>T] = 0	**False negative (FN)**	**True negative (TN)**

- Własność

$$\frac{\partial E(TP)}{\partial T} = -\frac{\partial E(FN)}{\partial T}$$

$$\frac{\partial E(FP)}{\partial T} = -\frac{\partial E(TN)}{\partial T}$$

  - Gdzie T nazywamy progiem odcięcia

Optymalizacja progu odcięcia

- Przypiszmy miarę efektu V(n) (np. mierzony pieniężnie koszt błędnej klasyfikacji) :

- Kryterium oceny modelu:
  - **oczekiwana wartość efektu**
- Cel

Klasa prawdziwa

Klasa prognozowana	1	0
[f(X)>T] = 1	V(TP)	V(FP)
[f(X)>T] = 0	V(FN)	V(TN)

$$V(TP) * E(TP) + V(FP) * E(FP) + V(FN) + E(FN) + V(TN) * E(TN) \rightarrow \max$$

- Optimum

$$\frac{\partial E(FN)}{\partial T} \Big/ \frac{\partial E(TN)}{\partial T} = (V(TN) - V(FP))/(V(TP) - V(FN))$$

- **Wybór T zależy od relatywnego kosztu błędu!**

Wybór odpowiedniej metryki

- Bardzo ważne jest też określenie jaka jest docelowa wartość błędu do której dążymy.
- W większości przypadków nie da się osiągnąć 100% trafności predykcji.

Błąd Bayesowski: $\int_{x \in H_1} P(C_0|x)p(x)dx + \int_{x \in H_0} P(C_1|x)p(x)dx$

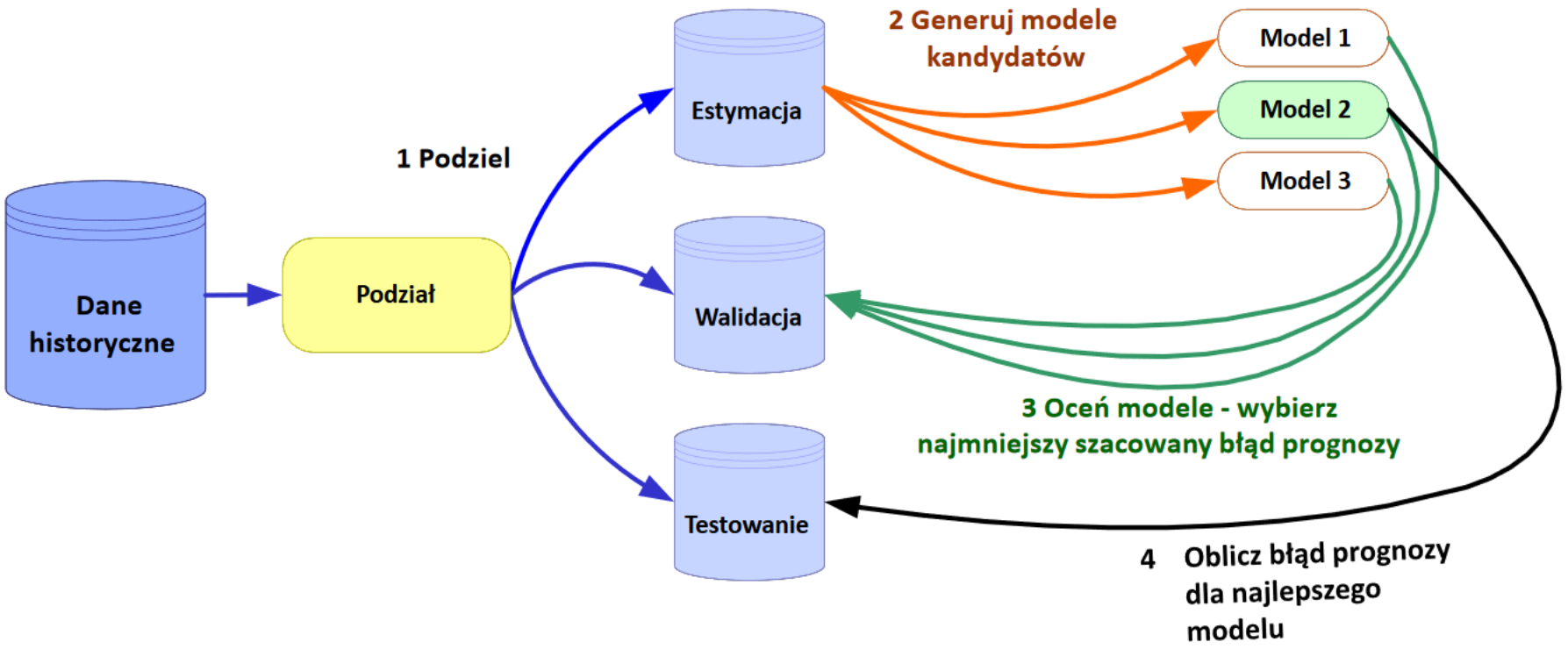
Wybór odpowiedniej metryki

- Określenie wielkości akceptowalnego błędu zależy od kilku podstawowych czynników:
  - Przede wszystkim od tego czy możliwe jest dalsze zbieranie danych (im więcej danych tym potencjalnie niższego błędu predykcji możemy się spodziewać).
  - Oraz od tego czy ze względu na czas i koszty możliwe jest tworzenie bardzo złożonych modeli.

Przygotowanie danych

- Przygotowując dane do uczenia modelu klasyfikacyjnego należy pamiętać o podzieleniu zbiorów na 3 części:
  - **Trenujący**
  - **Walidacyjny**
  - **Testowy**

Przygotowanie danych



Przygotowanie danych - skutki pominięcia zbioru walidacyjnego

- **Y**: binarna zmienna objaśniana
- **X1, X2, X3, X4**: losowe zmienne objaśniające niezależne między sobą; ze zmienną objaśnianą związana tylko zmienna X1
- 4 modele MNK oszacowane na zbiorze trenującym, za każdym razem dodawana kolejna jedna zmienna objaśniająca
- Wybrałem model o największej liczbie poprawnych klasyfikacji na zbiorze **uczącym**

Model	Uczący	Walidacyjny	Testowy
Stała+X1	64	64	63
Stała+X1-X2	65	62	63
Stała+X1-X3	65	62	62
Stała+X1-X4	66	62	61

Przygotowanie danych - skutki pominięcia zbioru testowego

- **Y**: binarna zmienna objaśniana
- **X1, X2, ... , X100**: losowe zmienne objaśniające niezależne między sobą i ze zmienną objaśnianą
- 100 modeli MNK oszacowane na zbiorze trenującym, za każdym razem wybierana jedna zmienna objaśniającą
- Wybrałem model o największej liczbie poprawnych klasyfikacji na zbiorze **walidacyjnym**
- Cztery najlepsze wyniki na zbiorze walidacyjnym (poprawny wynik to **50**)

Model	Uczący	Walidacyjny	Testowy
Najlepszy	57	66	56
Drugi	54	65	50
Trzeci	50	64	52
Trzeci	50	64	53

Przygotowanie danych

- Podział na 3 zbiory jest intuicyjnym i prostym sposobem poprawnego szacowania modeli klasyfikacyjnych.
- Przykładowy podział:
  - Zbiór trenujący: 60% obserwacji
  - Zbiór walidacyjny: 20% obserwacji
  - Zbiór testowy: 20% obserwacji
- Ma jednak zasadniczą wadę, tracimy dużą część obserwacji na których nie możemy trenować naszego modelu
- Jest to szczególnie ważne gdy zbieranie danych jest drogie i praco- lub czasochłonne.
- Dlatego często korzystamy z alternatywnych metod podziału:
  - Bootstrap aggregating (**bagging**)
  - Walidacja krzyżowa (**cross-validation**)

Bagging

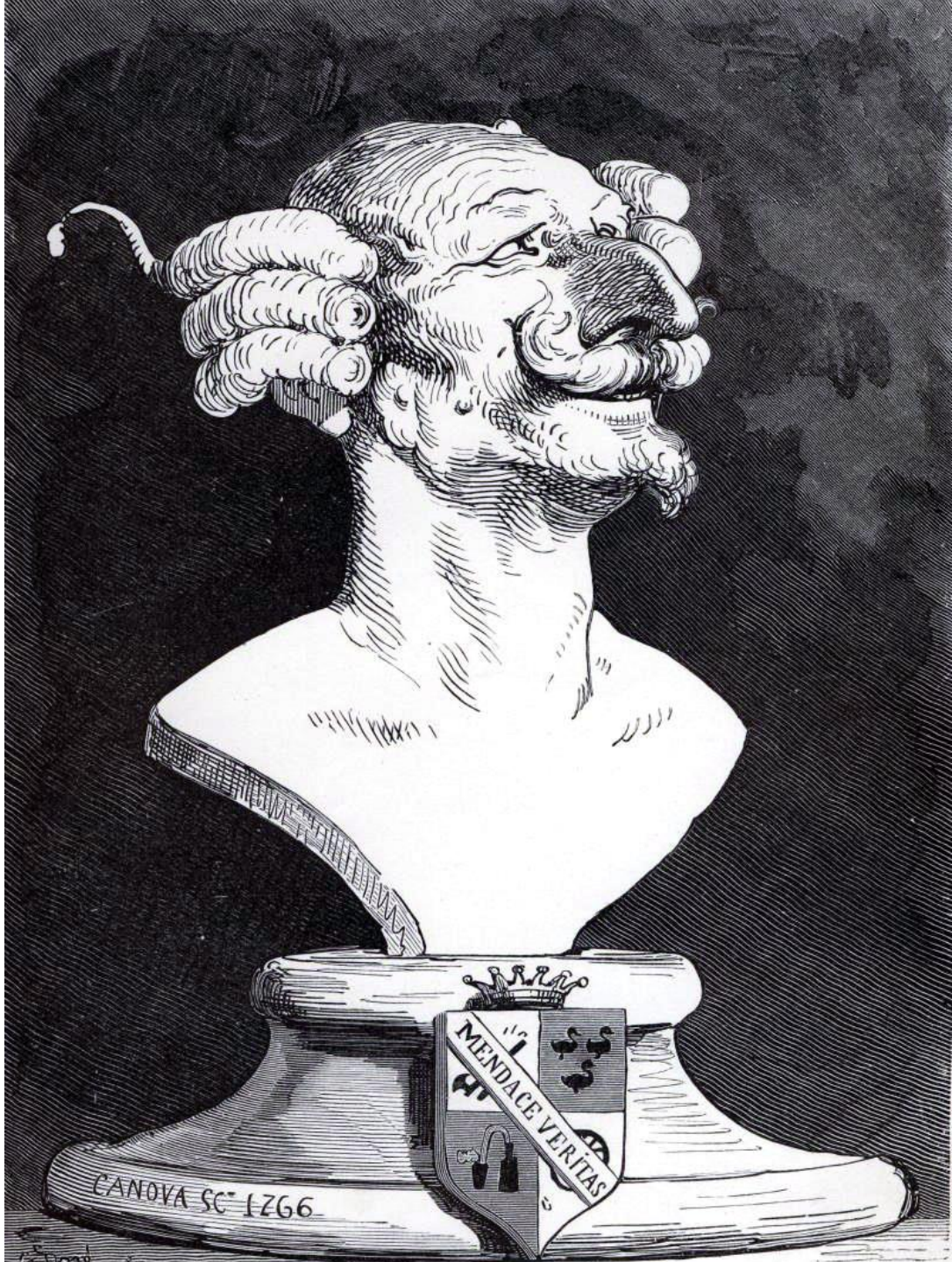
- Problem:
  - Mamy zebrane dane ale jest ich za mało żeby przeprowadzić poprawnie procedurę uczenia.
  - Co możemy zrobić?

Bagging

- Problem:
  - Mamy zebrane dane ale jest ich za mało żeby przeprowadzić poprawnie procedurę uczenia.
  - Co możemy zrobić?
- Bagging (Bootstrap aggregating) to prosta metoda, która pozwala nam obejść ten problem.

Bootstrapping

- Bootstrapping to prosta technika pozwalająca nam na „generowanie” nowych danych bazując na tych, które już zebraliśmy.
- W najprostszym ujęciu polega ona na tworzeniu nowych zbiorów danych poprzez losowanie ze zwracaniem próbek ze zbioru który już mamy:



Bootstrapping

- Bootstrapping to prosta technika pozwalająca nam na „generowanie” nowych danych bazując na tych, które już zebraliśmy.
- W najprostszym ujęciu polega ona na tworzeniu nowych zbiorów danych poprzez losowanie ze zwracaniem próbek ze zbioru który już mamy:

1 2 3 6 7 8 10 22 12 33

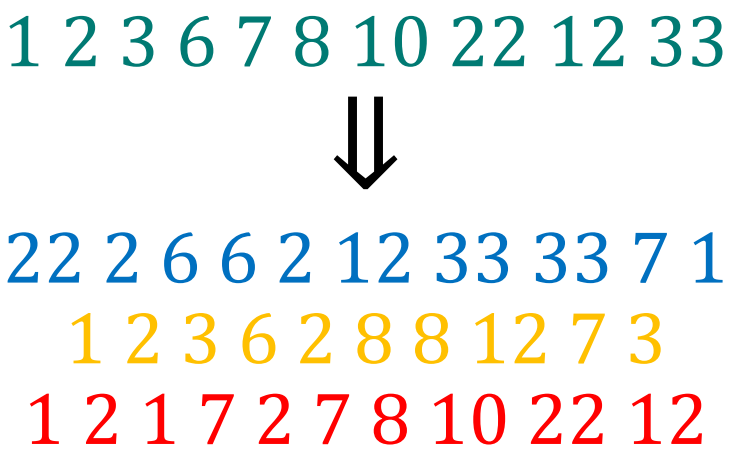
22 2 6 6 2 12 33 33 7 1

1 2 3 6 2 8 8 12 7 3

1 2 1 7 2 7 8 10 22 12

Bootstrapping

- Wygenerowane w ten sposób zmienne losowe (prawie) zachowują się jakby były **niezależne i o identycznym rozkładzie**.
- Co więcej, przy odpowiednio dużym zbiorze danych możemy być pewni, że unikalnych obserwacji będzie wystarczająco dużo żeby odwzorowanie danych i jakość uczenia nie były zachwiane.

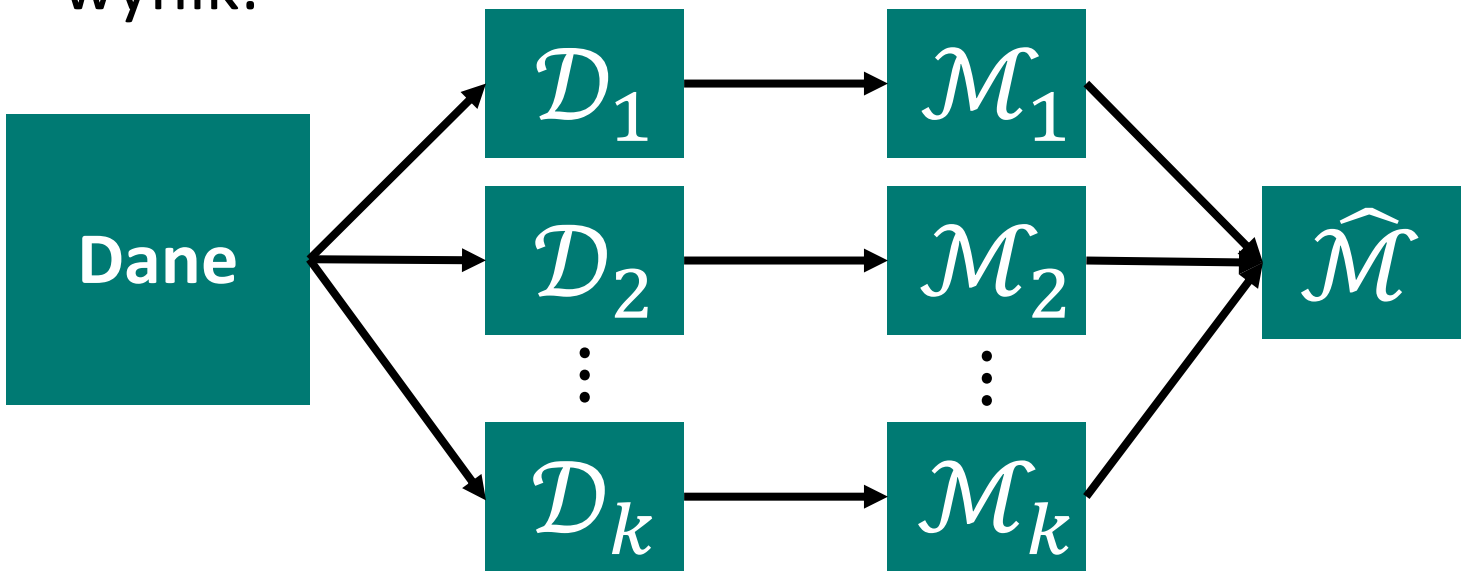


Bagging

- Procedura baggingu jako swoją bazę wykorzystuje właśnie bootstrappowane dane.
- Dodaje do nich jednak kolejny poziom (agregację).

Bagging

- Procedura baggingu jako swoją bazę wykorzystuje właśnie bootstrappowane dane.
- Dodaje do nich jednak kolejny poziom (agregację).
- Z n elementowego zbioru tworzymy k m elementowych zbiorów na których szacujemy k modeli. Naszą predykcją będzie ich uśredniony wynik:



Bagging

- Zaletą baggingu jest to, że szacujemy wiele modeli na różnych danych jednocześnie, dzięki czemu zmniejszamy wyraźnie wariancję końcowego oszacowania.
- Agregacja polega na:
  - Uśrednianiu wyników (w przypadku regresji):

$$\widehat{\mathcal{M}} = \frac{1}{k}\sum_{i=1}^{k}\mathcal{M}_i$$

  - Głosowaniu większościowym (w przypadku klasyfikacji):

$$\widehat{\mathcal{M}} = \underset{l}{\operatorname{argmax}}[|\mathcal{M}_i = l|]$$

Walidacja krzyżowa

- Walidacja krzyżowa (*cross-validation*) jest metodą pozwalającą na szacowanie i agregację wielu modeli bez konieczności tworzenia nowych zbiorów danych.
- W najpopularniejszym przypadku k-krotnej walidacji krzyżowej (*k-fold cross-validation*):

Walidacja krzyżowa

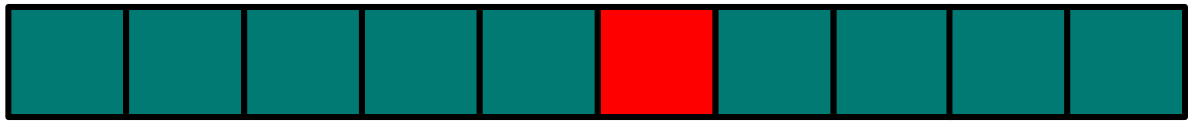
- Walidacja krzyżowa (*cross-validation*) jest metodą pozwalającą na szacowanie i agregację wielu modeli bez konieczności tworzenia nowych zbiorów danych.
- W najpopularniejszym przypadku k-krotnej walidacji krzyżowej (*k-fold cross-validation*):
  - Zaczynamy od losowego podzielenia n elementowego zbioru na k równych części.

Walidacja krzyżowa

- Walidacja krzyżowa (*cross-validation*) jest metodą pozwalającą na szacowanie i agregację wielu modeli bez konieczności tworzenia nowych zbiorów danych.
- W najpopularniejszym przypadku k-krotnej walidacji krzyżowej (*k-fold cross-validation*):
  - Zaczynamy od losowego podzielenia n elementowego zbioru na k równych części.
  - Następnie szacujemy model na $k - 1$ częściach, a jedną wykorzystujemy do wyznaczenia błędu oszacowania.

Walidacja krzyżowa

- Walidacja krzyżowa (*cross-validation*) jest metodą pozwalającą na szacowanie i agregację wielu modeli bez konieczności tworzenia nowych zbiorów danych.
- W najpopularniejszym przypadku k-krotnej walidacji krzyżowej (*k-fold cross-validation*):
  - Zaczynamy od losowego podzielenia n elementowego zbioru na k równych części.
  - Następnie szacujemy model na $k - 1$ częściach, a jedną wykorzystujemy do wyznaczenia błędu oszacowania.
  - Procedurę powtarzamy dopóki każda z części nie zostanie raz wykorzystana do oceny modelu:



Walidacja krzyżowa

- Walidacja krzyżowa (*cross-validation*) jest metodą pozwalającą na szacowanie i agregację wielu modeli bez konieczności tworzenia nowych zbiorów danych.
- W najpopularniejszym przypadku k-krotnej walidacji krzyżowej (*k-fold cross-validation*):
  - Zaczynamy od losowego podzielenia n elementowego zbioru na k równych części.
  - Następnie szacujemy model na $k - 1$ częściach, a jedną wykorzystujemy do wyznaczenia błędu oszacowania.
  - Procedurę powtarzamy dopóki każda z części nie zostanie raz wykorzystana do oceny modelu:

Walidacja krzyżowa

- Walidacja krzyżowa (*cross-validation*) jest metodą pozwalającą na szacowanie i agregację wielu modeli bez konieczności tworzenia nowych zbiorów danych.
- W najpopularniejszym przypadku k-krotnej walidacji krzyżowej (*k-fold cross-validation*):
  - Zaczynamy od losowego podzielenia n elementowego zbioru na k równych części.
  - Następnie szacujemy model na $k-1$ częściach, a jedną wykorzystujemy do wyznaczenia błędu oszacowania.
  - Procedurę powtarzamy dopóki każda z części nie zostanie raz wykorzystana do oceny modelu:

Walidacja krzyżowa

- Metody walidacji krzyżowej możemy podzielić na:
  - Wyczerpujące:
    - *Leave-one-out*
    - *Leave-p-out*
  - Niewyczerpujące:
    - *k-krotna walidacja krzyżowa*
    - *Monte Carlo cross-validation*
- Walidacja krzyżowa może też być stratyfikowana jeżeli wymaga tego od nas struktura zbioru danych.