Statystyczna Eksploracja Danych

Wykład 3 - metoda najbliższych sąsiadów, ocena skuteczności klasyfikatora

dr inż. Julian Sienkiewicz

7 marca 2019

Metoda najbliższych sąsiadów

- bezpośrednie oszacowanie warunkowego prawdopodobieństwa a posteriori $p(j|\mathbf{x})$ przynależności zaobserwowanej wartości $\mathbf{x}$ do klasy j,

Metoda najbliższych sąsiadów

- bezpośrednie oszacowanie warunkowego prawdopodobieństwa a posteriori $p(j|\mathbf{x})$ przynależności zaobserwowanej wartości $\mathbf{x}$ do klasy j,
- każdy estymator musi się opierać na próbie uczącej:

Metoda najbliższych sąsiadów

- bezpośrednie oszacowanie warunkowego prawdopodobieństwa a posteriori $p(j|\mathbf{x})$ przynależności zaobserwowanej wartości $\mathbf{x}$ do klasy j,
- każdy estymator musi się opierać na próbie uczącej:
  - naturalny sposób oszacowania $p(j|\mathbf{x})$ to porównanie gęstości rozmieszczenia obserwacji z różnych klas w **bezpośrednim otoczeniu** $\mathbf{x}$.

Metoda najbliższych sąsiadów

- bezpośrednie oszacowanie warunkowego prawdopodobieństwa a posteriori $p(j|\mathbf{x})$ przynależności zaobserwowanej wartości $\mathbf{x}$ do klasy j,
- każdy estymator musi się opierać na próbie uczącej:
  - naturalny sposób oszacowania $p(j|\mathbf{x})$ to porównanie gęstości rozmieszczenia obserwacji z różnych klas w **bezpośrednim otoczeniu** $\mathbf{x}$.

Bezpośrednie otoczenie

- Tylko czym jest **bezpośrednie otoczenie**?

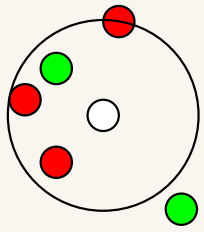
Metoda najbliższych sąsiadów

- bezpośrednie oszacowanie warunkowego prawdopodobieństwa a posteriori $p(j|\mathbf{x})$ przynależności zaobserwowanej wartości $\mathbf{x}$ do klasy j,
- każdy estymator musi się opierać na próbie uczącej:
  - naturalny sposób oszacowania $p(j|\mathbf{x})$ to porównanie gęstości rozmieszczenia obserwacji z różnych klas w **bezpośrednim otoczeniu** $\mathbf{x}$.

Bezpośrednie otoczenie

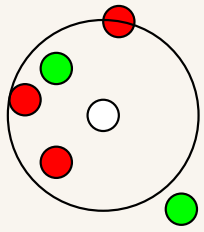
- Tylko czym jest **bezpośrednie otoczenie**?
- Jak je zdefiniować? Ile obserwacji z każdej z klas g jest w bezpośrednim otoczeniu?

Metoda najbliższych sąsiadów



k obserwacji (k-nearest neighbors, k-nn) najbliższych $\mathbf{x}_i$ spośród wszystkich $\mathbf{x}_1, \ldots, \mathbf{x}_n$

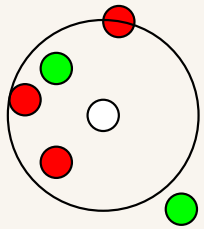
Metoda najbliższych sąsiadów



k obserwacji (k-nearest neighbors, k-nn) najbliższych $\mathbf{x}_i$ spośród wszystkich $\mathbf{x}_1, \ldots, \mathbf{x}_n$

Czyli **bezpośrednim otoczeniem** punktu $\mathbf{x}$ jest kula w R^p, o środku w $\mathbf{x}$ i promieniu takim, żeby znalazło się w nim dokładnie k obserwacji $\mathbf{x}_i$

Metoda najbliższych sąsiadów



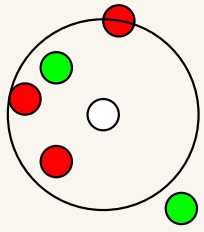
k obserwacji (k-nearest neighbors, k-nn) najbliższych $\mathbf{x}_i$ spośród wszystkich $\mathbf{x}_1, \ldots, \mathbf{x}_n$

Czyli **bezpośrednim otoczeniem** punktu $\mathbf{x}$ jest kula w R^p, o środku w $\mathbf{x}$ i promieniu takim, żeby znalazło się w nim dokładnie k obserwacji $\mathbf{x}_i$

$$p(j|\mathbf{x}) = \frac{\sum_{\langle k \rangle} \mathbf{x} \in j}{k}$$

Obserwacja $\mathbf{x}$ zostanie sklasyfikowana do tej klasy j, z której pochodzi najwięcej spośród k najbliższych punktowi $\mathbf{x}$ obserwacji próby uczącej

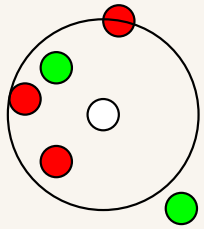
Wariant k-nn: (k,l)-nn



Podobnie, jak poprzednio, ale wśród *k* najbliższych sąsiadów przynajmniej *l* należy do klasy [tutaj np. (4,3)-nn)]

W przeciwnym wypadku decyzja nie zostaje podjęta (dobre przy nierównych kosztach klasyfikacji)

Wariant k-nn: (k,l)-nn

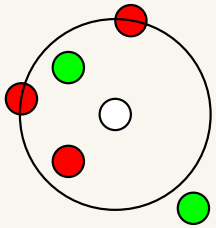


Podobnie, jak poprzednio, ale wśród k najbliższych sąsiadów przynajmniej l należy do klasy [tutaj np. (4,3)-nn)]

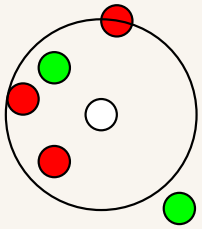
W przeciwnym wypadku decyzja nie zostaje podjęta (dobre przy nierównych kosztach klasyfikacji)

Uwagi

1. Jeżeli jest więcej niż k równoodległych obserwacji, to bierzemy wszystkie (tutaj: 3-nn, ale bierzemy 4 punkty)



Wariant k-nn: (k,l)-nn

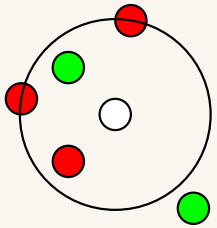


Podobnie, jak poprzednio, ale wśród k najbliższych sąsiadów przynajmniej l należy do klasy [tutaj np. (4,3)-nn)]

W przeciwnym wypadku decyzja nie zostaje podjęta (dobre przy nierównych kosztach klasyfikacji)

Uwagi

1. Jeżeli jest więcej niż k równoodległych obserwacji, to bierzemy wszystkie (tutaj: 3-nn, ale bierzemy 4 punkty)

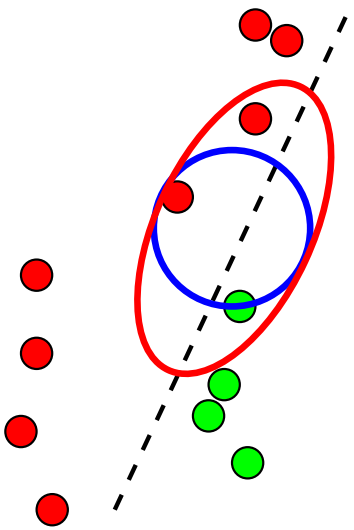


2. w przypadku “remisu” decyzja należy do eksperymentatora

Odległość w metodzie najbliższych sąsiadów

Kolosalne znaczenie dla jakości dyskryminacji ma postać przyjętej metryki (definicji odległości):

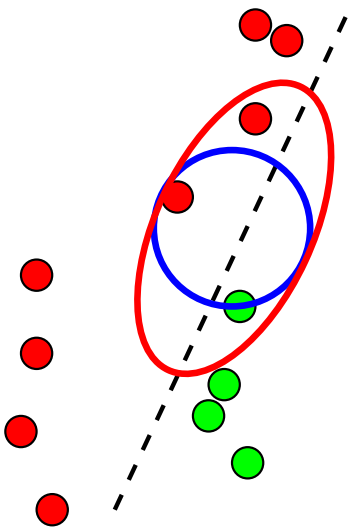
- najczęściej: euklidesowa,
- niekiedy: odległość Mahalanobisa
  $d_m(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\mathbf{A}(\mathbf{x} - \mathbf{y})^T}$



Odległość w metodzie najbliższych sąsiadów

Kolosalne znaczenie dla jakości dyskryminacji ma postać przyjętej metryki (definicji odległości):

- najczęściej: euklidesowa,
- niekiedy: odległość Mahalanobisa
  $d_m(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\mathbf{A}(\mathbf{x} - \mathbf{y})^T}$



- szczególnie istotne na granicy klas
- można dobrać macierz **A** tak, aby zminimalizować błędy (za pomocą kroswalidacji)

Odległość Mahalanobisa

Odległość między dwoma punktami w n-wymiarowej przestrzeni, która różnicuje wkład poszczególnych składowych oraz wykorzystuje korelacje miedzy nimi.

$$d_m(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\mathbf{C}^{-1}(\mathbf{x} - \mathbf{y})^T} \tag{1}$$

Przypadki:

- równe wariancje i brak korelacji ($\mathbf{C} = \mathbf{I}$) (odl. euklidesowa, punkty o równej odległości tworzą okrąg),

Odległość Mahalanobisa

Odległość między dwoma punktami w n-wymiarowej przestrzeni, która różnicuje wkład poszczególnych składowych oraz wykorzystuje korelacje miedzy nimi.

$$d_m(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\mathbf{C}^{-1}(\mathbf{x} - \mathbf{y})^T} \tag{1}$$

Przypadki:

- równe wariancje i brak korelacji ($\mathbf{C} = \mathbf{I}$) (odl. euklidesowa, punkty o równej odległości tworzą okrąg),
- różne wariancje i brak korelacji $\mathbf{C} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$ (punkty o równej odległości tworzą elipsę)

Odległość Mahalanobisa

Odległość między dwoma punktami w n-wymiarowej przestrzeni, która różnicuje wkład poszczególnych składowych oraz wykorzystuje korelacje miedzy nimi.

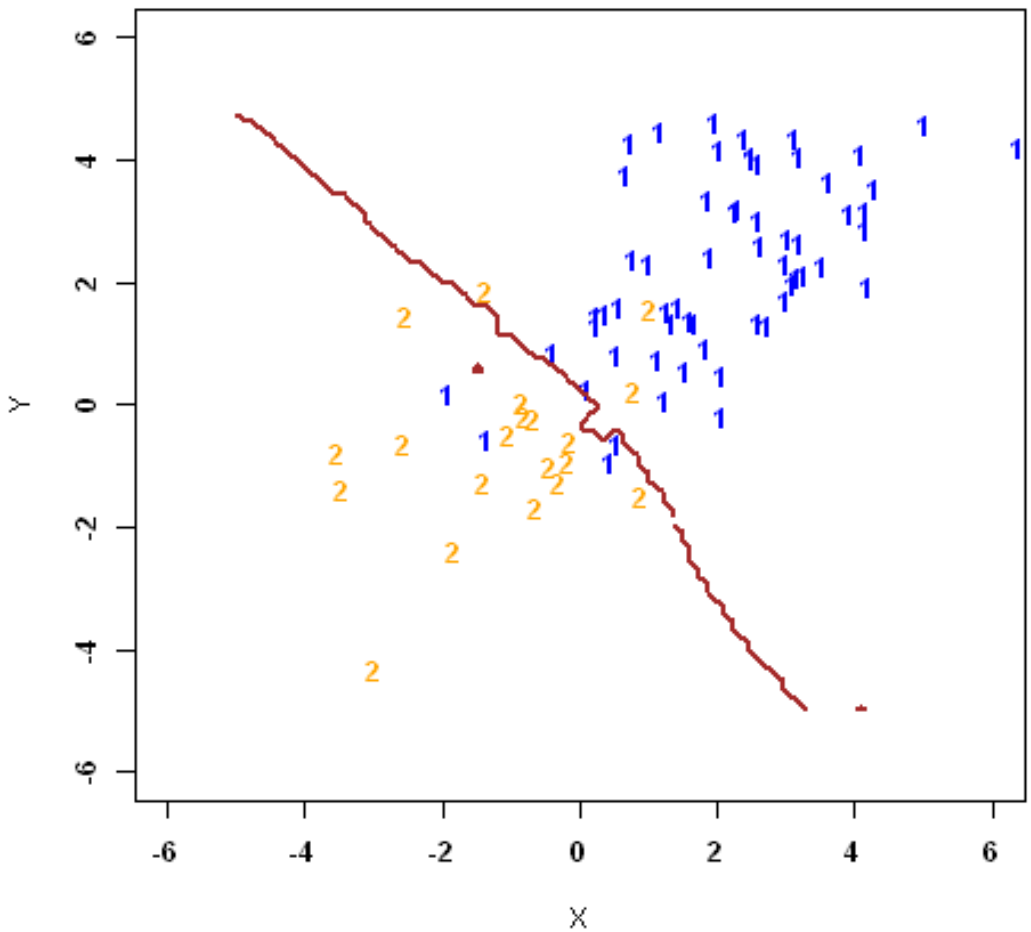
$$d_m(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\mathbf{C}^{-1}(\mathbf{x} - \mathbf{y})^T} \tag{1}$$

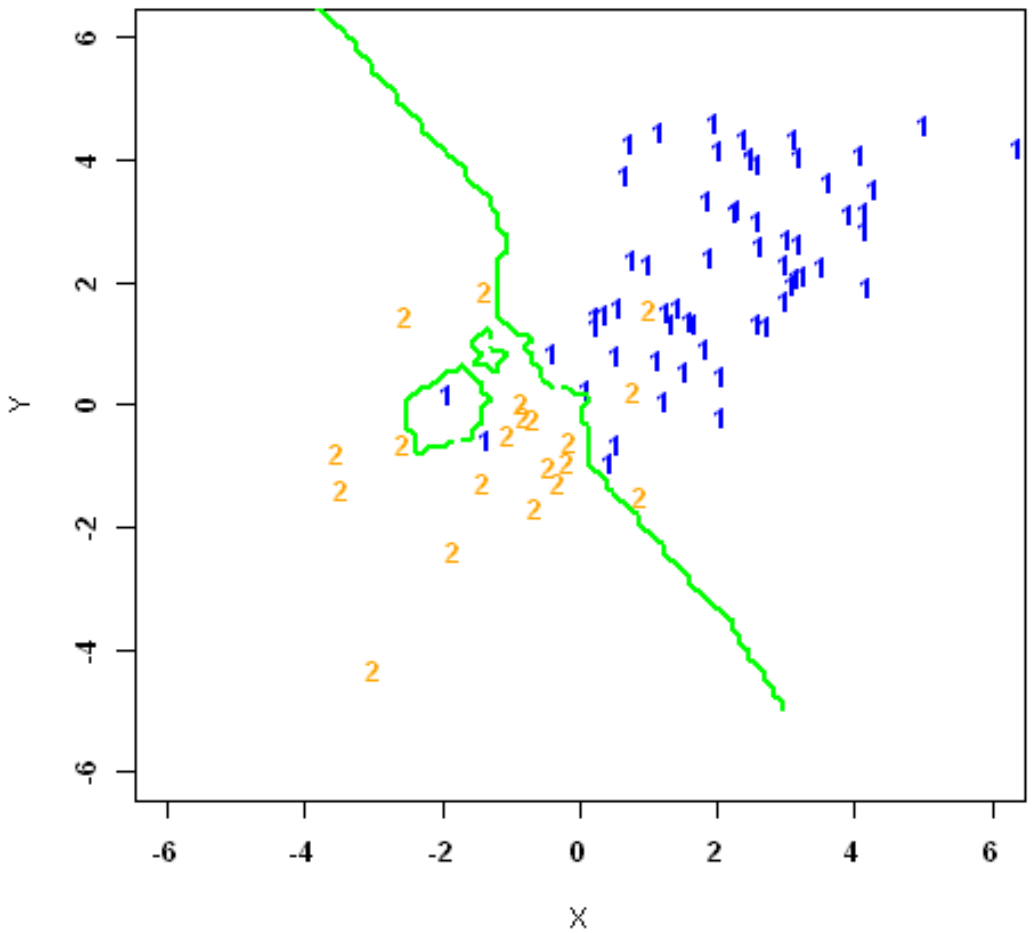
Przypadki:

- równe wariancje i brak korelacji ($\mathbf{C} = \mathbf{I}$) (odl. euklidesowa, punkty o równej odległości tworzą okrąg),
- różne wariancje i brak korelacji $\mathbf{C} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$ (punkty o równej odległości tworzą elipsę)
- różne wariancje i korelacje $\mathbf{C}$ - pełna macierz kowariancji (punkty o równej odległości tworzą obrócona elipsę)

Przykład

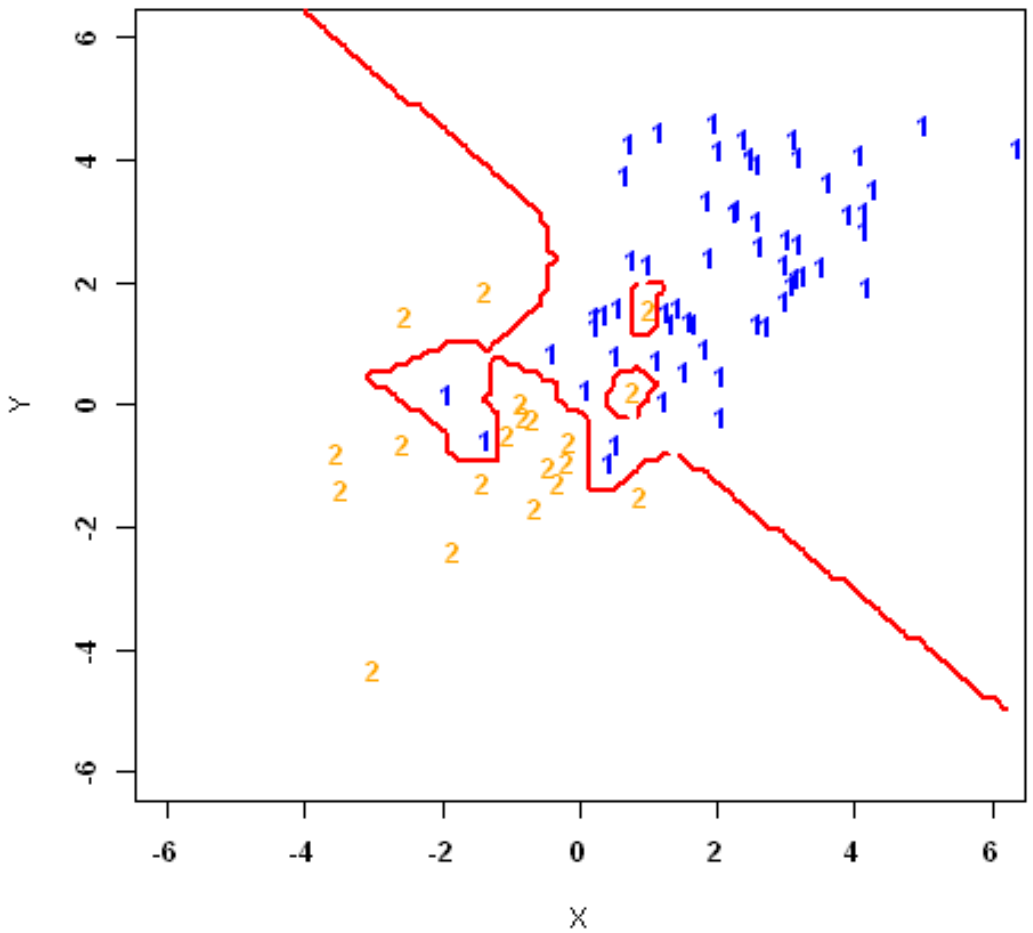
- 5-nn, $\alpha = 0.91$





Przykład

- 5-nn, $\alpha = 0.91$
- 3-nn, $\alpha = 0.95$



Przykład

- 5-nn, $\alpha = 0.91$
- 3-nn, $\alpha = 0.95$
- 1-nn, $\alpha = 1.00$

Przekleństwo wymiarowości (ang. *dimensionality curse*)

- metoda k-nn ma duże problemy w przypadku danych o wielu wymiarach,

Przekleństwo wymiarowości (ang. *dimensionality curse*)

- metoda k-nn ma duże problemy w przypadku danych o wielu wymiarach,
- weźmy przypadek, gdy punkty są rozłożone jednorodnie w kostce o wymiarach $[-\frac{1}{2}, \frac{1}{2}]^p$,

Przekleństwo wymiarowości (ang. *dimensionality curse*)

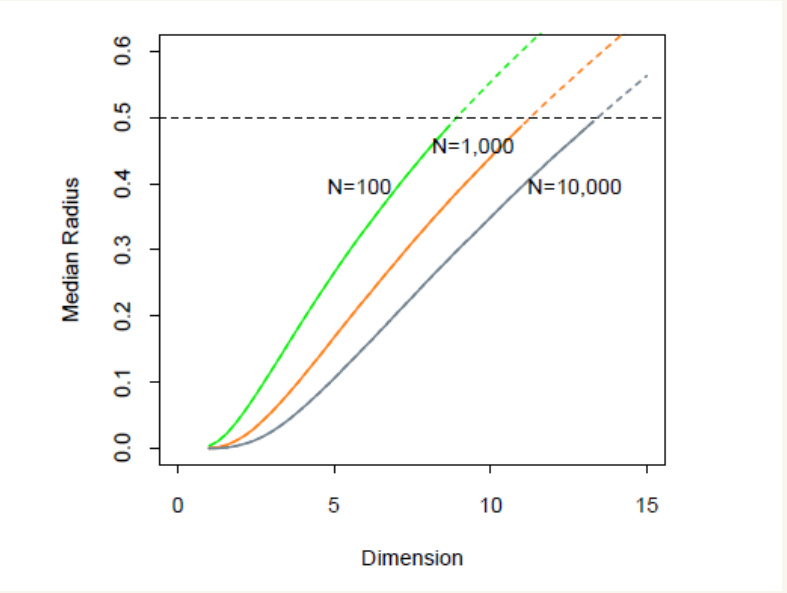
- metoda k-nn ma duże problemy w przypadku danych o wielu wymiarach,
- weźmy przypadek, gdy punkty są rozłożone jednorodnie w kostce o wymiarach $[-\frac{1}{2}, \frac{1}{2}]^p$,
- poszukajmy mediany promienia otoczenia dla 1-nn zlokalizowanego w środku układu współrzędnych,

Przekleństwo wymiarowości (ang. *dimensionality curse*)

- metoda k-nn ma duże problemy w przypadku danych o wielu wymiarach,
- weźmy przypadek, gdy punkty są rozłożone jednorodnie w kostce o wymiarach $[-\frac{1}{2}, \frac{1}{2}]^p$,
- poszukajmy mediany promienia otoczenia dla 1-nn zlokalizowanego w środku układu współrzędnych,
- mediana szybko zbliża się do wartości 1/2.

Przekleństwo wymiarowości (ang. *dimensionality curse*)

- metoda k-nn ma duże problemy w przypadku danych o wielu wymiarach,
- weźmy przypadek, gdy punkty są rozłożone jednorodnie w kostce o wymiarach $[-\frac{1}{2}, \frac{1}{2}]^p$,
- poszukajmy mediany promienia otoczenia dla 1-nn zlokalizowanego w środku układu współrzędnych,
- mediana szybko zbliża się do wartości 1/2.



Przekleństwo wymiarowości (ang. *dimensionality curse*)

- jak sobie z tym poradzić?

Przekleństwo wymiarowości (ang. *dimensionality curse*)

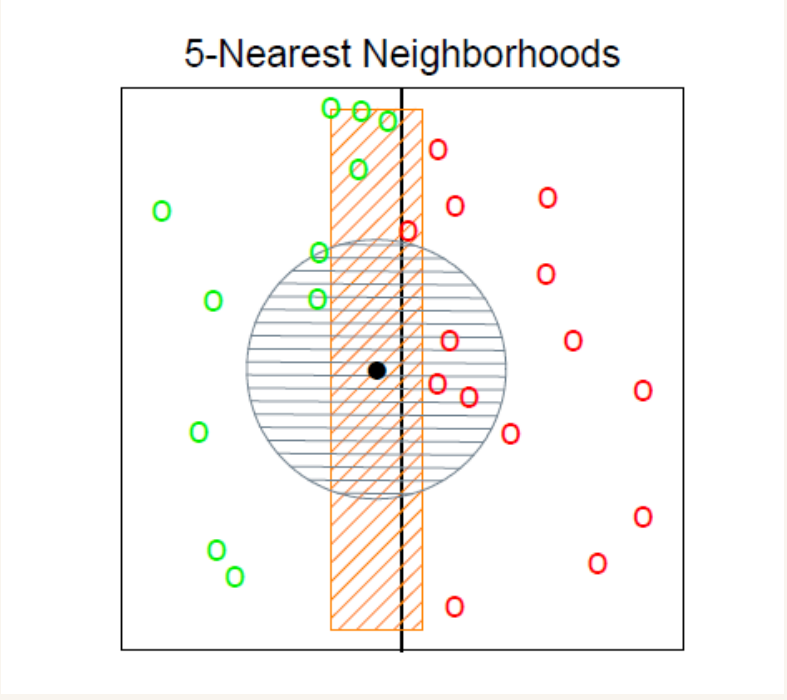
- jak sobie z tym poradzić?
- potrzebna jest dodatkowa informacja o układzie, dotycząca np. rozrzutu (wariancji) punktów w konkretnym wymiarze,

Przekleństwo wymiarowości (ang. *dimensionality curse*)

- jak sobie z tym poradzić?
- potrzebna jest dodatkowa informacja o układzie, dotycząca np. rozrzutu (wariancji) punktów w konkretnym wymiarze,
- wykorzystywane są wtedy metody wstępne (np. LDA, PCA), aby zredukować zbiór danych

Przekleństwo wymiarowości (ang. *dimensionality curse*)

- jak sobie z tym poradzić?
- potrzebna jest dodatkowa informacja o układzie, dotycząca np. rozrzutu (wariancji) punktów w konkretnym wymiarze,
- wykorzystywane są wtedy metody wstępne (np. LDA, PCA), aby zredukować zbiór danych



Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

Mając zbiór $X = \{x_1, x_2, ..., x_n\}$ punktów z PU:

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

Mając zbiór $X = \{x_1, x_2, ..., x_n\}$ punktów z PU:

- dodajemy do zbioru U punkt x_1,

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

Mając zbiór $X = \{x_1, x_2, ..., x_n\}$ punktów z PU:

- dodajemy do zbioru U punkt x_1,
- przeszukując zbiór X, szukamy elementu zbioru x, którego najbliższy prototyp z U ma inną klasę niż niż x,

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

Mając zbiór $X = \{x_1, x_2, ..., x_n\}$ punktów z PU:

- dodajemy do zbioru U punkt x_1,
- przeszukując zbiór X, szukamy elementu zbioru x, którego najbliższy prototyp z U ma inną klasę niż niż x,
- usuwamy x z X i dodajmy do U,

Condensed Nearest Neighbors (CNN)

W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

Mając zbiór $X = \{x_1, x_2, ..., x_n\}$ punktów z PU:

- dodajemy do zbioru U punkt x_1,
- przeszukując zbiór X, szukamy elementu zbioru x, którego najbliższy prototyp z U ma inną klasę niż niż x,
- usuwamy x z X i dodajmy do U,,
- wykonujemy do momentu, gdy nie znajdziemy już nowych elementów do dodania do U,

Condensed Nearest Neighbors (CNN)

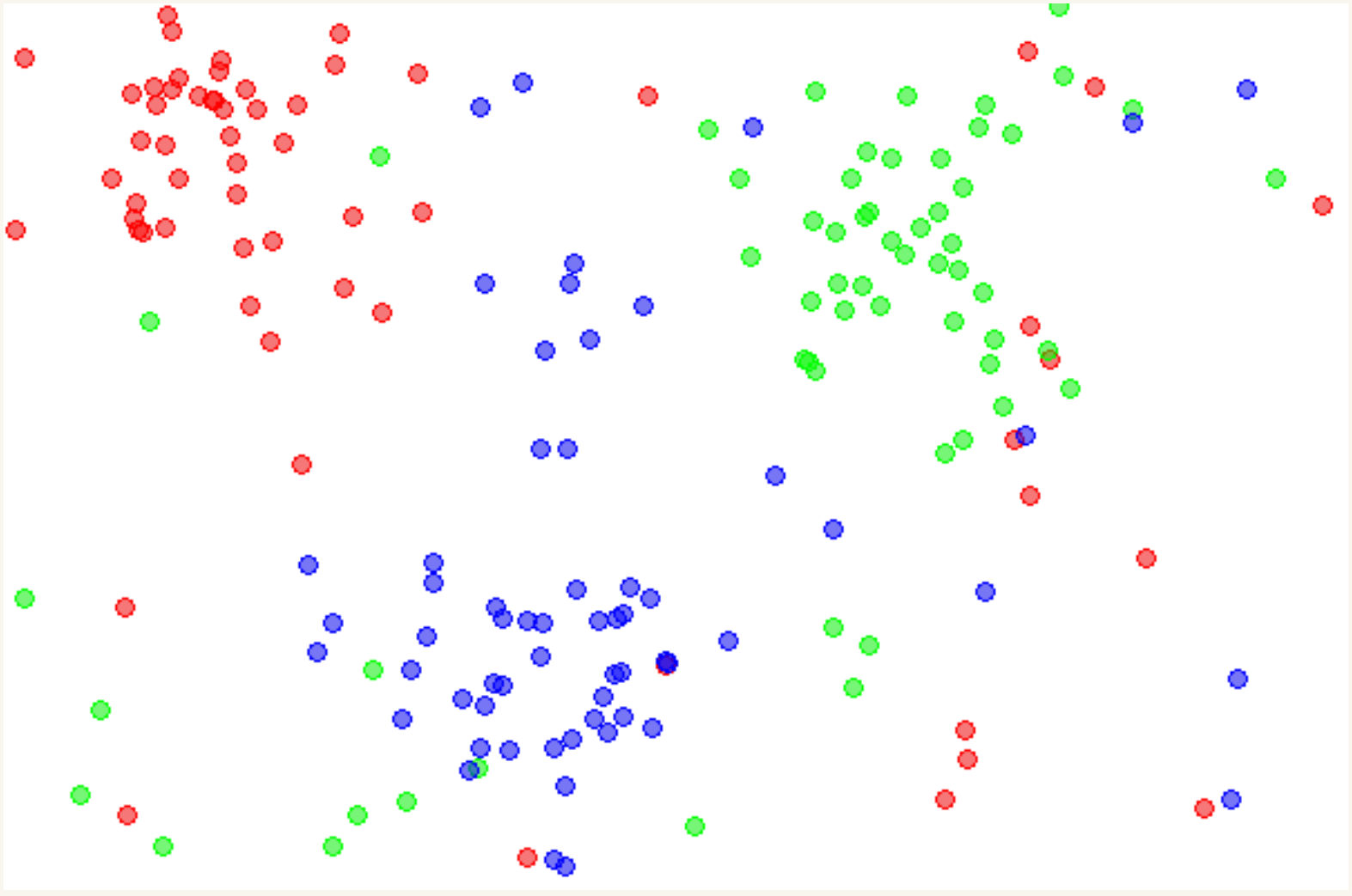
W wielu przypadkach konieczne jest ograniczenie zbioru danych, na podstawie których podejmowane są decyzje (zbyt wiele odległości do uwzględnienia).

- po pierwsze, pozbywamy się punktów odstających (outliers): czyli tych, które zostały źle sklasyfikowane dla danej metody,
- pozostałe punkty rozdzielamy na **prototypy** oraz punkty **absorbujące**,

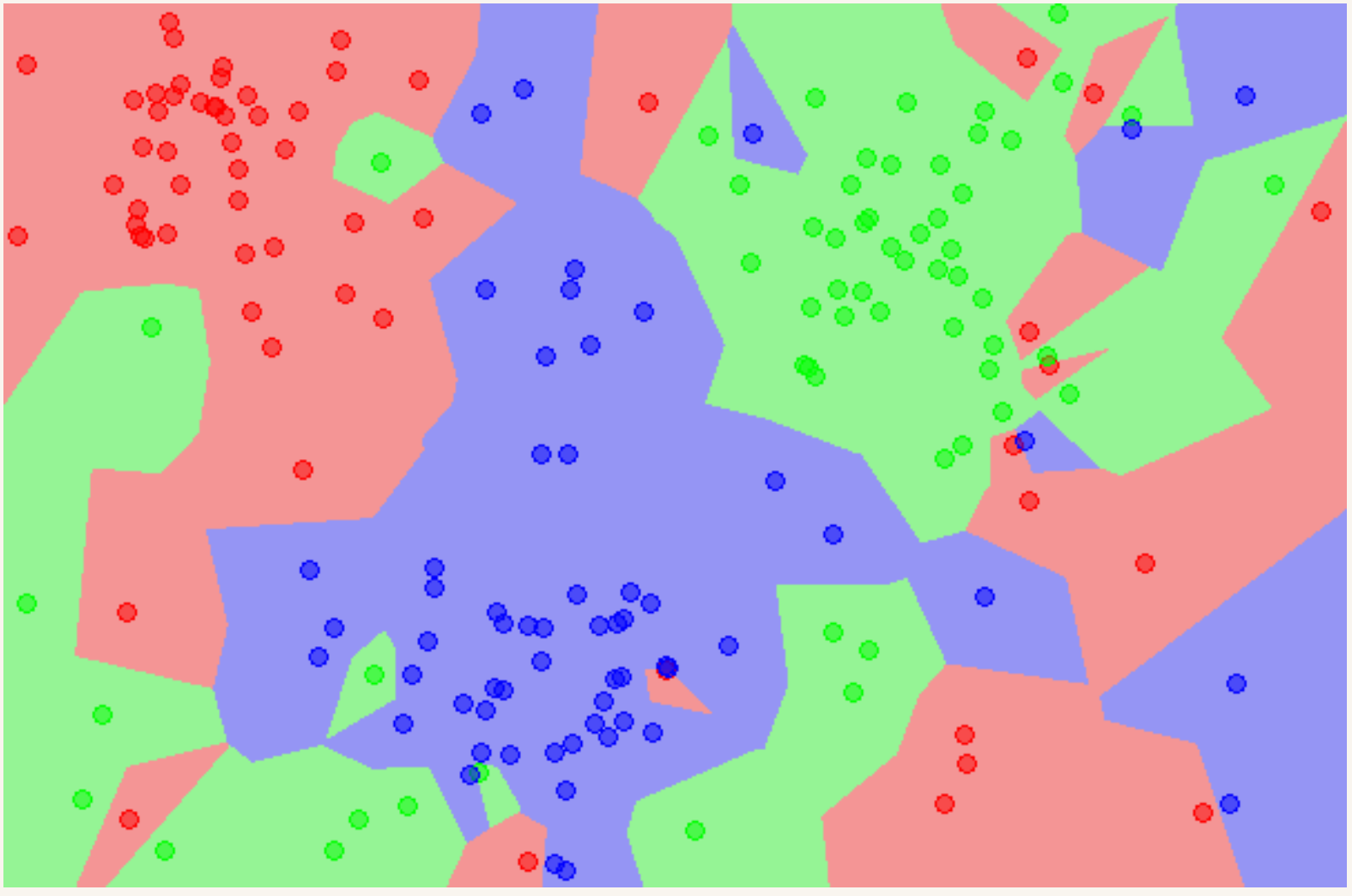
Mając zbiór $X = \{x_1, x_2, ..., x_n\}$ punktów z PU:

- dodajemy do zbioru U punkt x_1,
- przeszukując zbiór X, szukamy elementu zbioru x, którego najbliższy prototyp z U ma inną klasę niż niż x,
- usuwamy x z X i dodajmy do U,,
- wykonujemy do momentu, gdy nie znajdziemy już nowych elementów do dodania do U,,

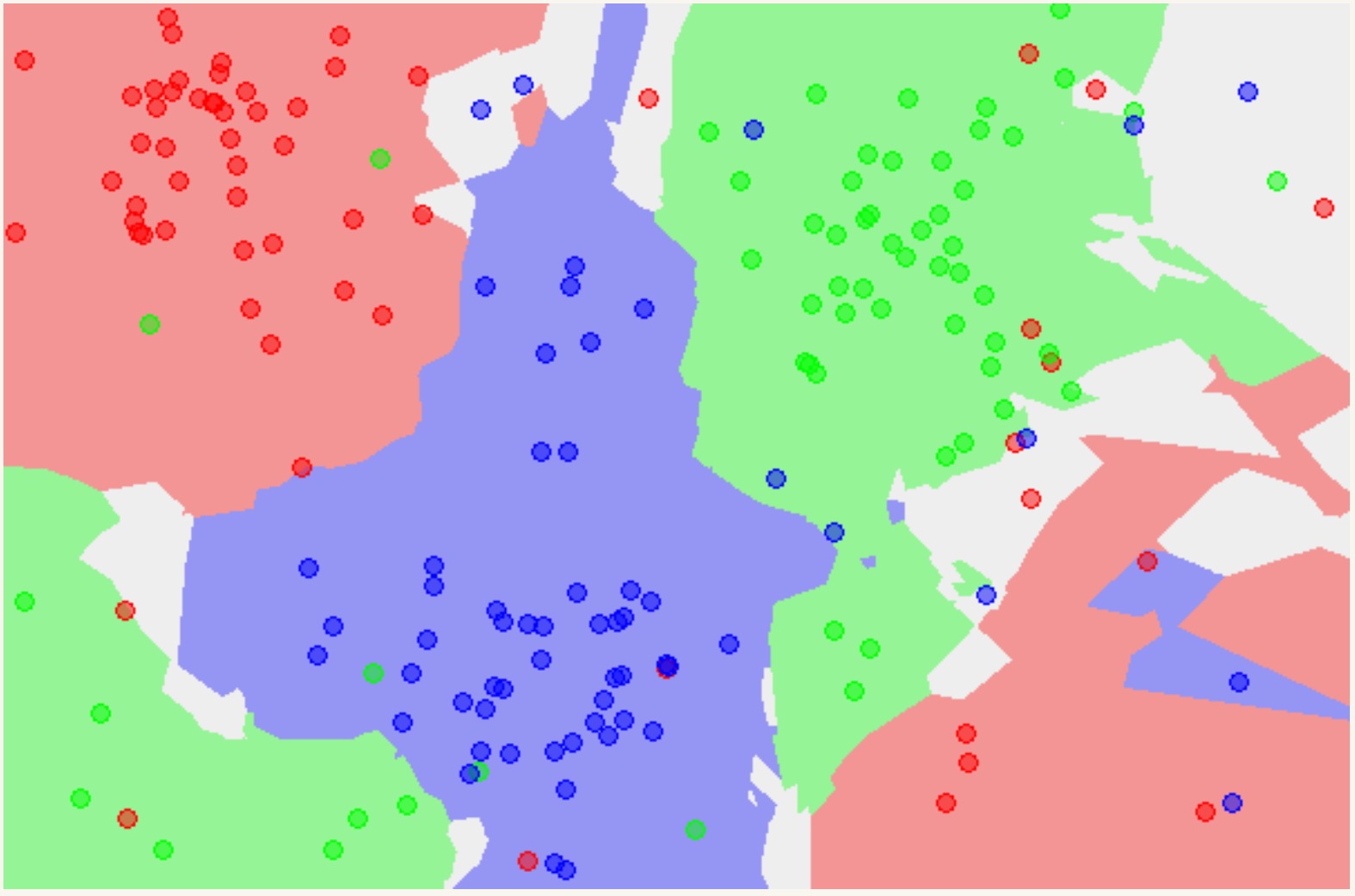
Wyniki



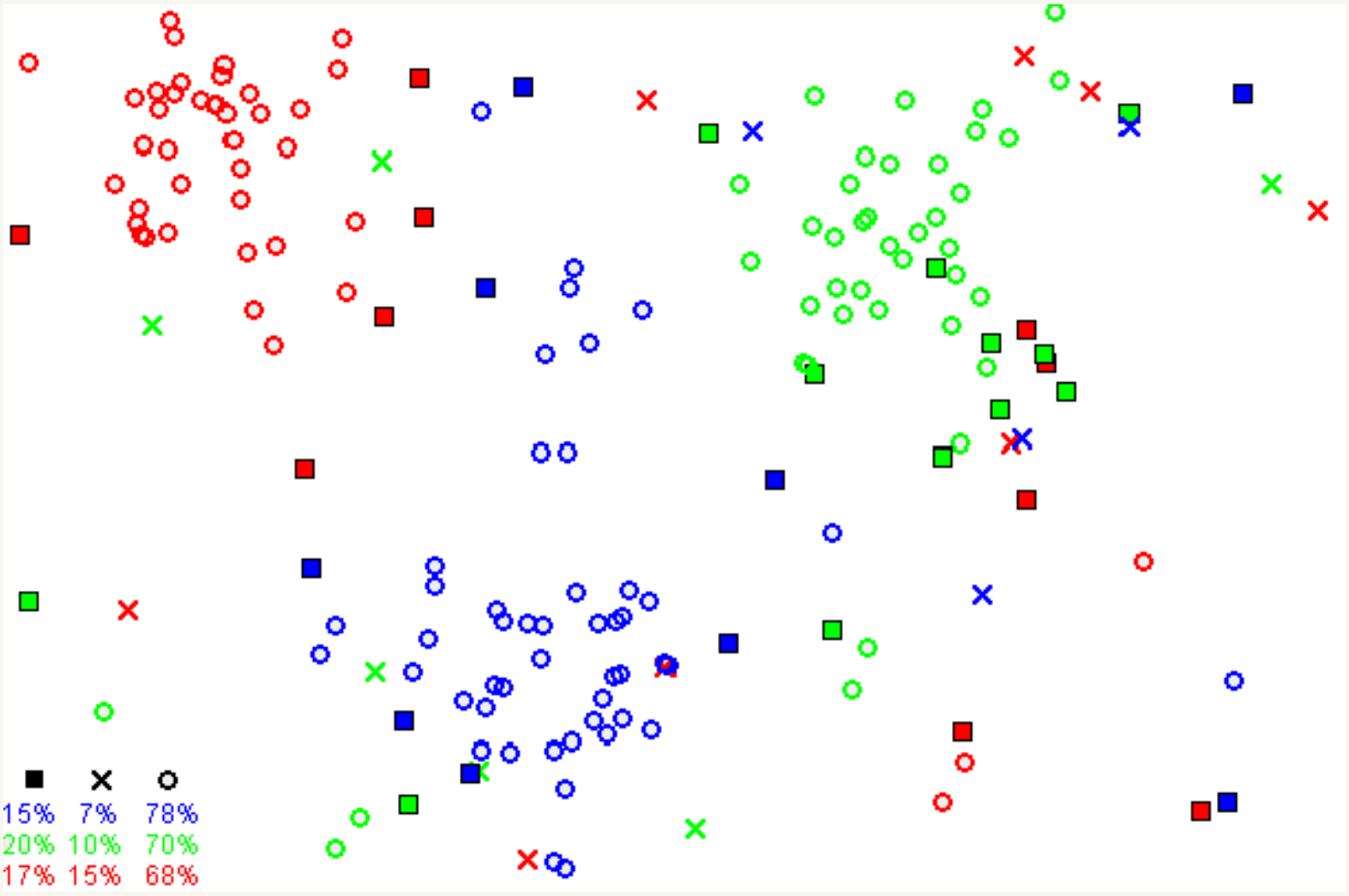
Wyniki



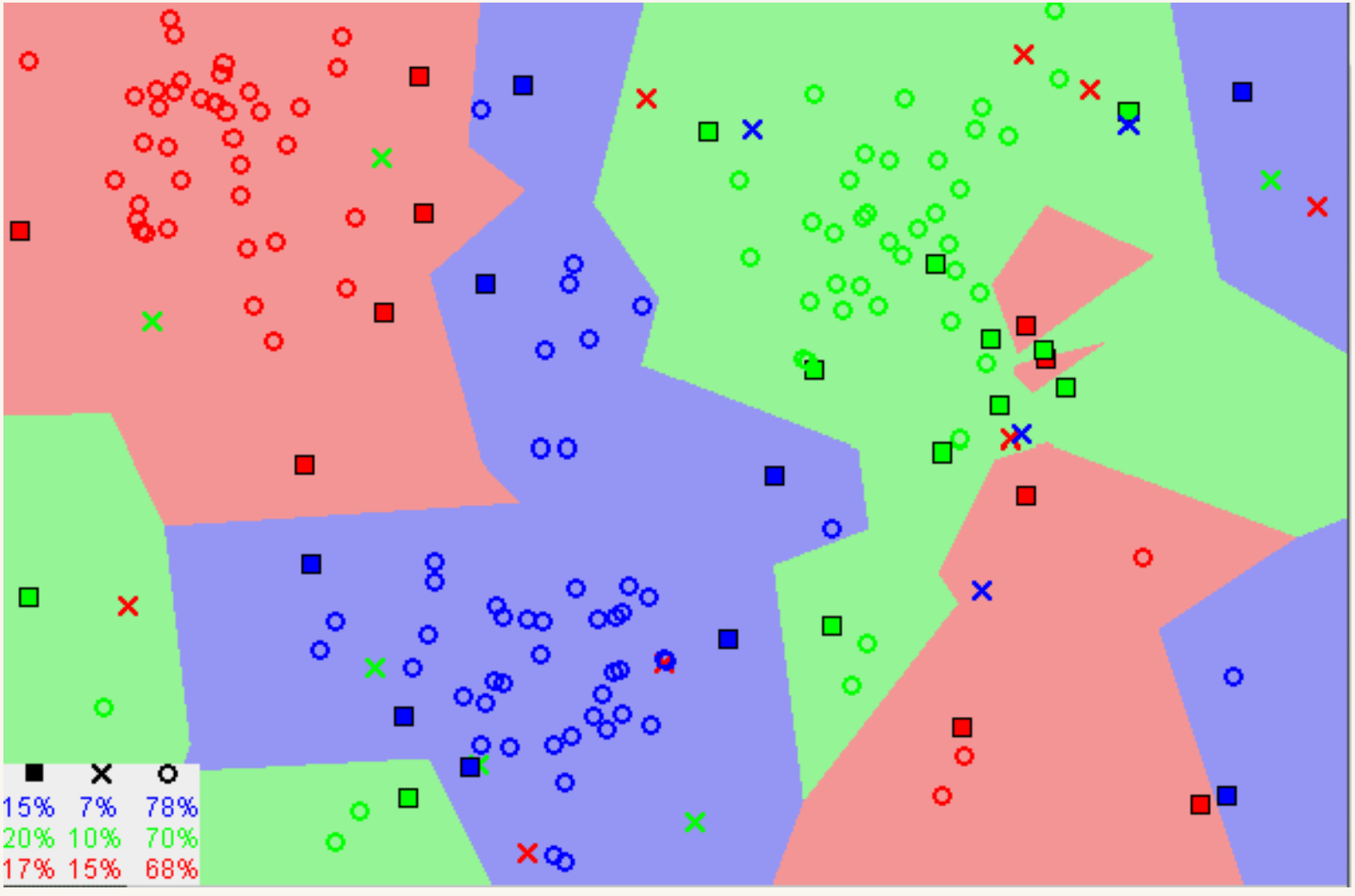
Wyniki



Wyniki



Wyniki



Interesuje nas rozwiązanie zadania klasyfikacji pod nadzorem

Interesuje nas rozwiązanie zadania klasyfikacji pod nadzorem

- mamy **PRÓBĘ UCZĄCĄ (PU)**, na której Mozela skonstruować różne klasyfikatory (wiemy, jaka jest przynależność obserwacji do klas),

Interesuje nas rozwiązanie zadania klasyfikacji pod nadzorem

- mamy **PRÓBĘ UCZĄCĄ (PU)**, na której Mozela skonstruować różne klasyfikatory (wiemy, jaka jest przynależność obserwacji do klas),
- zakładamy, że dysponujemy też inną losową próbą obserwacji (o znanych klasach), ale niezależną od próby uczącej = **PRÓBA WALIDACYJNA (PW)**,

Interesuje nas rozwiązanie zadania klasyfikacji pod nadzorem

- mamy **PRÓBĘ UCZĄCĄ (PU)**, na której Mozela skonstruować różne klasyfikatory (wiemy, jaka jest przynależność obserwacji do klas),
- zakładamy, że dysponujemy też inną losową próbą obserwacji (o znanych klasach), ale niezależną od próby uczącej = **PRÓBA WALIDACYJNA (PW)**,
- wtedy oszacowanie prawdopodobieństwa błędnej klasyfikacji

Interesuje nas rozwiązanie zadania klasyfikacji pod nadzorem

- mamy **PRÓBĘ UCZĄCĄ (PU)**, na której Mozela skonstruować różne klasyfikatory (wiemy, jaka jest przynależność obserwacji do klas),
- zakładamy, że dysponujemy też inną losową próbą obserwacji (o znanych klasach), ale niezależną od próby uczącej = **PRÓBA WALIDACYJNA (PW)**,
- wtedy oszacowanie prawdopodobieństwa błędnej klasyfikacji

procent błędnych klasyfikacji dokonanych na PW

Interesuje nas rozwiązanie zadania klasyfikacji pod nadzorem

- mamy **PRÓBĘ UCZĄCĄ (PU)**, na której Mozela skonstruować różne klasyfikatory (wiemy, jaka jest przynależność obserwacji do klas),
- zakładamy, że dysponujemy też inną losową próbą obserwacji (o znanych klasach), ale niezależną od próby uczącej = **PRÓBA WALIDACYJNA (PW)**,
- wtedy oszacowanie prawdopodobieństwa błędnej klasyfikacji

procent błędnych klasyfikacji dokonanych na PW

- stąd wybieramy klasyfikator, który popełnił najmniej błędów.

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

- w przeciwnym razie otrzymamy **obciążone (zaniżone)** oszacowanie błędu, gdyż konstrukcja klasyfikatora opiera się na dopasowaniu do PU

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

- w przeciwnym razie otrzymamy **obciążone (zaniżone)** oszacowanie błędu, gdyż konstrukcja klasyfikatora opiera się na dopasowaniu do PU

Pozostaje kwestia ostatecznej oceny prawdopodobieństwa dokonania błędnej oceny

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

- w przeciwnym razie otrzymamy **obciążone (zaniżone)** oszacowanie błędu, gdyż konstrukcja klasyfikatora opiera się na dopasowaniu do PU

Pozostaje kwestia ostatecznej oceny prawdopodobieństwa dokonania błędnej oceny

- **nie** powinno to się odbywać na podstawie PW $\rightarrow$ tu już wybieramy klasyfikator,

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

- w przeciwnym razie otrzymamy **obciążone (zaniżone)** oszacowanie błędu, gdyż konstrukcja klasyfikatora opiera się na dopasowaniu do PU

Pozostaje kwestia ostatecznej oceny prawdopodobieństwa dokonania błędnej oceny

- **nie** powinno to się odbywać na podstawie PW $\rightarrow$ tu już wybieramy klasyfikator,
- potrzebujemy kolejnej, niezależnej od poprzednich **PRÓBY TESTOWEJ (PT)**

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

- w przeciwnym razie otrzymamy **obciążone (zaniżone)** oszacowanie błędu, gdyż konstrukcja klasyfikatora opiera się na dopasowaniu do PU

Pozostaje kwestia ostatecznej oceny prawdopodobieństwa dokonania błędnej oceny

- **nie** powinno to się odbywać na podstawie PW $\rightarrow$ tu już wybieramy klasyfikator,
- potrzebujemy kolejnej, niezależnej od poprzednich **PRÓBY TESTOWEJ (PT)**
- PT nie jest potrzebna, jeżeli na podstawie PU budujemy tylko jeden klasyfikator $\rightarrow$ wystarczy wtedy tylko PW.

PRÓBA WALIDACYJNA musi być niezależna od PRÓBY UCZĄCEJ (PW $\neq$ PU) !

- w przeciwnym razie otrzymamy **obciążone (zaniżone)** oszacowanie błędu, gdyż konstrukcja klasyfikatora opiera się na dopasowaniu do PU

Pozostaje kwestia ostatecznej oceny prawdopodobieństwa dokonania błędnej oceny

- **nie** powinno to się odbywać na podstawie PW $\rightarrow$ tu już wybieramy klasyfikator,
- potrzebujemy kolejnej, niezależnej od poprzednich **PRÓBY TESTOWEJ (PT)**
- PT nie jest potrzebna, jeżeli na podstawie PU budujemy tylko jeden klasyfikator $\rightarrow$ wystarczy wtedy tylko PW.

Podział danych (PU/PW/PT): 50/25/25 (lub 60/20/20) $\rightarrow$ brak dobrej odpowiedzi

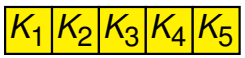
Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy?

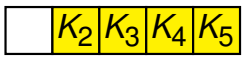
Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy? $\rightarrow$ Stosujemy **korswalidację** (sprawdzanie krzyżowe), polegającą na wielokrotnym wykorzystaniu PU, tak zorganizowanego, aby obciążenie było możliwie małe.

Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy? $\rightarrow$ Stosujemy **korswalidację** (sprawdzanie krzyżowe), polegającą na wielokrotnym wykorzystaniu PU, tak zorganizowanego, aby obciążenie było możliwie małe.

- PU zostaje podzielona na K (np. $K = 5$) części 

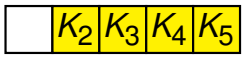
Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy? $\rightarrow$ Stosujemy **korswalidację** (sprawdzanie krzyżowe), polegającą na wielokrotnym wykorzystaniu PU, tak zorganizowanego, aby obciążenie było możliwie małe.

- PU zostaje podzielona na K (np. $K = 5$) części 

- tworzy się K różnych **pseudoprób**, powstałych poprzez usuwanie z próby oryginalnej jednej z K części 

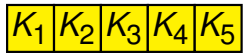
Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy? $\rightarrow$ Stosujemy **korswalidację** (sprawdzanie krzyżowe), polegającą na wielokrotnym wykorzystaniu PU, tak zorganizowanego, aby obciążenie było możliwie małe.

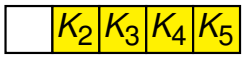
- PU zostaje podzielona na K (np. $K = 5$) części 

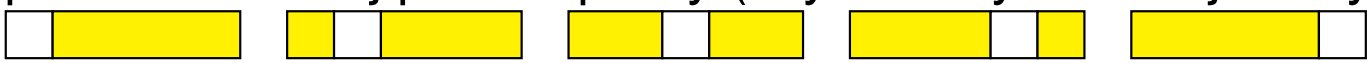
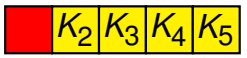
- tworzy się K różnych **pseudoprób**, powstałych poprzez usuwanie z próby oryginalnej jednej z K części 

- klasyfikator tworzony jest K-krotnie, za każdym razem na podstawie innej pseudopróby (czyli mamy K wersji klasyfikatora)

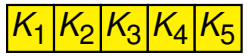
Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy? $\rightarrow$ Stosujemy **korswalidację** (sprawdzanie krzyżowe), polegającą na wielokrotnym wykorzystaniu PU, tak zorganizowanego, aby obciążenie było możliwie małe.

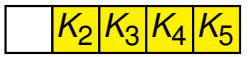
- PU zostaje podzielona na K (np. $K = 5$) części 

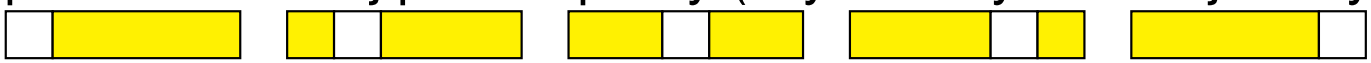
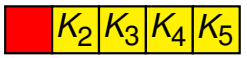
- tworzy się K różnych **pseudoprób**, powstałych poprzez usuwanie z próby oryginalnej jednej z K części 

- klasyfikator tworzony jest K-krotnie, za każdym razem na podstawie innej pseudopróby (czyli mamy K wersji klasyfikatora) 
- każda K-ta wersja klasyfikatora jest oceniana poprzez sprawdzenie liczby błędnych klasyfikacji na tej części oryginalnej próby, która **nie weszła** do K-tej pseudopróby 

Często ze względu na niedobór danych trzeba zrezygnować z wydzielenia PW i PT. Co wtedy? $\rightarrow$ Stosujemy **korswalidację** (sprawdzanie krzyżowe), polegającą na wielokrotnym wykorzystaniu PU, tak zorganizowanego, aby obciążenie było możliwie małe.

- PU zostaje podzielona na K (np. $K = 5$) części 

- tworzy się K różnych **pseudoprób**, powstałych poprzez usuwanie z próby oryginalnej jednej z K części 

- klasyfikator tworzony jest K-krotnie, za każdym razem na podstawie innej pseudopróby (czyli mamy K wersji klasyfikatora) 
- każda K-ta wersja klasyfikatora jest oceniana poprzez sprawdzenie liczby błędnych klasyfikacji na tej części oryginalnej próby, która **nie weszła** do K-tej pseudopróby 

Korswalidacja (cd):

- Σ wszystkich błędnych klasyfikacji / liczebność PU = prawdop. błędnej klasyfikacji,

Korswalidacja (cd):

- Σ wszystkich błędnych klasyfikacji / liczebność PU = prawdop. błędnej klasyfikacji,
- po wybraniu klasyfikatora konstruuje się go raz jeszcze $\rightarrow$ tym razem na podstawie **całej PU**,

Korswalidacja (cd):

- Σ wszystkich błędnych klasyfikacji / liczebność PU = prawdop. błędnej klasyfikacji,
- po wybraniu klasyfikatora konstruuje się go raz jeszcze $\rightarrow$ tym razem na podstawie **całej PU**,
- najczęściej wybiera się $K = 5$ lub $K = 10$,

Korswalidacja (cd):

- Σ wszystkich błędnych klasyfikacji / liczebność PU = prawdop. błędnej klasyfikacji,
- po wybraniu klasyfikatora konstruuje się go raz jeszcze $\rightarrow$ tym razem na podstawie **całej PU**,
- najczęściej wybiera się $K = 5$ lub $K = 10$,
- często wykorzystuje się także ***n*-krotną kroswalidację** (ang. *leave-one-out cross-validation*):
  - każda pseudopróba powstaje poprzez usunięcie tylko jednej obserwacji (czyli liczność próby to $n - 1$),
  - każda wersja klasyfikatora oceniana jest na podstawie klasyfikowania **jednej obserwacji**.

Równie często wykorzystywanym sposobem radzenia sobie z problemem małych ilości danych jest metoda **bootstrap**:

Równie często wykorzystywanym sposobem radzenia sobie z problemem małych ilości danych jest metoda **bootstrap**:

- dokonanie wielokrotnego **repróbkowania** elementów z PU,

Równie często wykorzystywanym sposobem radzenia sobie z problemem małych ilości danych jest metoda **bootstrap**:

- dokonanie wielokrotnego **repróbkowania** elementów z PU,
- losowanie **ze zwracaniem** z PU o liczności n (np. PU={1,2,3,4,5} $\rightarrow$ {1,1,2,5,5})
- tworzymy psedudopróby (np. 1000),
- praktycznie żadna pseudopróba nie zawiera wszystkich elementów PU, średnio prawdopodobieństwo niewylosowania elementu PU to $\left(1-\frac{1}{n}\right)^n \rightarrow \mathrm{e}^{-1} \approx 0.368$, czyli ok. 1/3 nie zostaje wylosowana

Równie często wykorzystywanym sposobem radzenia sobie z problemem małych ilości danych jest metoda **bootstrap**:

- dokonanie wielokrotnego **repróbkowania** elementów z PU,
- losowanie **ze zwracaniem** z PU o liczności n (np. PU={1,2,3,4,5} $\rightarrow$ {1,1,2,5,5})
- tworzymy psedudopróby (np. 1000),
- praktycznie żadna pseudopróba nie zawiera wszystkich elementów PU, średnio prawdopodobieństwo niewylosowania elementu PU to $\left(1-\frac{1}{n}\right)^n \rightarrow \mathrm{e}^{-1} \approx 0.368$, czyli ok. 1/3 nie zostaje wylosowana
- korzystając z wylosowanych n-elementowych pseudoprób konstruujemy kolejne wersje tego samego klasyfikatora
  1. dla każdego elementu PU oblicza się ułamek błędnych klasyfikacji tego elementu przez wszystkie wersje klasyfikatora, do których konstrukcji **nie użyto** tego elemntu,
  2. oblicza się średnią wartość ułamków dla wszystkich elementów PU

Dwie praktyczne uwagi

1. należy zauważyć, że milcząco przyjmujemy założenie, iż rozkłady (rozkłady w klasach, prawdopodobieństwa a priori) przyszłych obserwacji są takie same, jak w PU
2. ostatecznie można przecież zrezygnować z PW i PT, korswalidacji czy bootstrap i zbudować klasyfikator na PU i oceniać też na PU $\rightarrow$ **POWTÓRNE PODSTAWIENIE**, wykorzystywane tylko wtedy, gdy mamy bardzo prostą postać reguły dyskryminacyjnej

Koszty błędnej klasyfikacji mogą czasem zależeć do jakiej klasy należy dana obserwacja:

- test diagnostyczny orzekający chorobę - koszt(błędnie, że chory) $<$ koszt(błędnie, że zdrów)

Koszty błędnej klasyfikacji mogą czasem zależeć do jakiej klasy należy dana obserwacja:

- test diagnostyczny orzekający chorobę - koszt(błędnie, że chory) $<$ koszt(błędnie, że zdrów)
- ocena zdolności kredytowej - koszt(błędnie, że niezdolny) $<$ koszt(błędnie, że zdolny),

Koszty błędnej klasyfikacji mogą czasem zależeć do jakiej klasy należy dana obserwacja:

- test diagnostyczny orzekający chorobę - koszt(błędnie, że chory) $<$ koszt(błędnie, że zdrów)
- ocena zdolności kredytowej - koszt(błędnie, że niezdolny) $<$ koszt(błędnie, że zdolny),

Czyli nie tylko chcemy mieć możliwie mało błędnych sklasyfikowań, ale jeśli jakieś muszą się pojawić, to lepiej, żeby byli to zdrowi niż chorzy.

Koszty błędnej klasyfikacji mogą czasem zależeć do jakiej klasy należy dana obserwacja:

- test diagnostyczny orzekający chorobę - koszt(błędnie, że chory) $<$ koszt(błędnie, że zdrów)
- ocena zdolności kredytowej - koszt(błędnie, że niezdolny) $<$ koszt(błędnie, że zdolny),

Czyli nie tylko chcemy mieć możliwie mało błędnych sklasyfikowań, ale jeśli jakieś muszą się pojawić, to lepiej, żeby byli to zdrowi niż chorzy.
W ocenie błędów dla poszczególnych klas pomocna jest **macierz pomyłek** (ang. *confusion matrix*)

	faktycznie CHORZY	faktycznie ZDROWI
sklasyfikowani jako CHORZY	TRUE POSITIVE (**TN**)	FALSE POSITIVE (**FP**)
sklasyfikowani jako ZDROWI	FALSE NEGATIVE (**FN**)	TRUE NEGATIVE (**TP**)

Przykład: mamy test medyczny przeprowadzony na próbie $n = 300$ osób, w której $n_1 = 100$ to faktycznie chorzy, a $n_2 = 200$ to faktycznie zdrowi.

Przykład: mamy test medyczny przeprowadzony na próbie $n = 300$ osób, w której $n_1 = 100$ to faktycznie chorzy, a $n_2 = 200$ to faktycznie zdrowi.

Macierz pomyłek

	faktycznie CHORZY	faktycznie ZDROWI
sklasyfikowani jako CHORZY	**97**	**24**
sklasyfikowani jako ZDROWI	**3**	**176**

Przykład: mamy test medyczny przeprowadzony na próbie $n = 300$ osób, w której $n_1 = 100$ to faktycznie chorzy, a $n_2 = 200$ to faktycznie zdrowi.

Macierz pomyłek

	faktycznie CHORZY	faktycznie ZDROWI
sklasyfikowani jako CHORZY	**97**	**24**
sklasyfikowani jako ZDROWI	**3**	**176**

Skuteczność (*accuracy*) $ACC = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{FP}+\mathrm{FN}+\mathrm{TP}+\mathrm{TN}} = \frac{273}{300} = 0.91$

Przykład: mamy test medyczny przeprowadzony na próbie $n = 300$ osób, w której $n_1 = 100$ to faktycznie chorzy, a $n_2 = 200$ to faktycznie zdrowi.

Macierz pomyłek

	faktycznie CHORZY	faktycznie ZDROWI
sklasyfikowani jako CHORZY	**97**	**24**
sklasyfikowani jako ZDROWI	**3**	**176**

Skuteczność (*accuracy*) $ACC = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{FP}+\mathrm{FN}+\mathrm{TP}+\mathrm{TN}} = \frac{273}{300} = 0.91$

$$R = \frac{TP}{TP+FN} = TPR = 0.97$$

czułość (*sensitivity, recall, TPR - true positive ratio*)

Daje oszacowanie prawdopodobieństwa przewidzenia przez test choroby, pod warunkiem, że pacjent jest chory

$$S = \frac{TN}{TN+FP} = TNR = 0.88$$

specyficzność (*specifity, true negative ratio*)

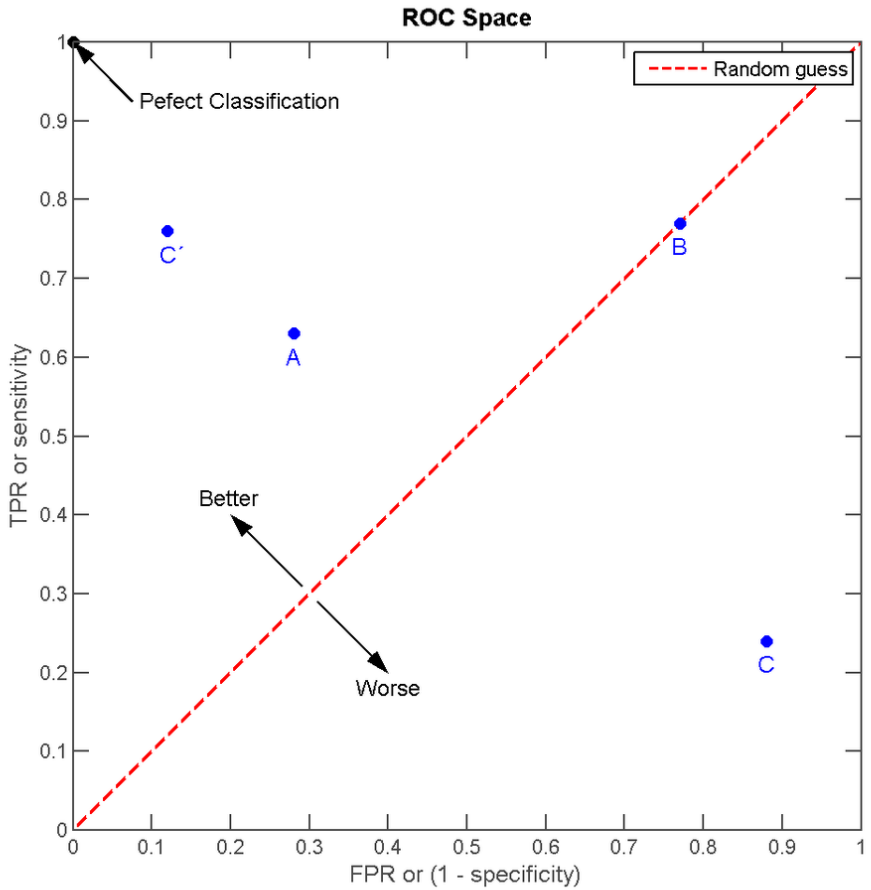
Daje oszacowanie prawdopodobieństwa przewidzenia przez test, że pacjent jest zdrowy, pod warunkiem, że nie cierpi na chorobę

Krzywa ROC

Poszczególne klasyfikatory można porównać graficznie w tzw. **przestrzeni ROC** (*Receiver Operating Characteristics*). W tym celu na osi *X* odkładamy wartość 1 − *TNR* (inaczej *FPR* - false positive ratio), a na osi *Y* - *TPR*.

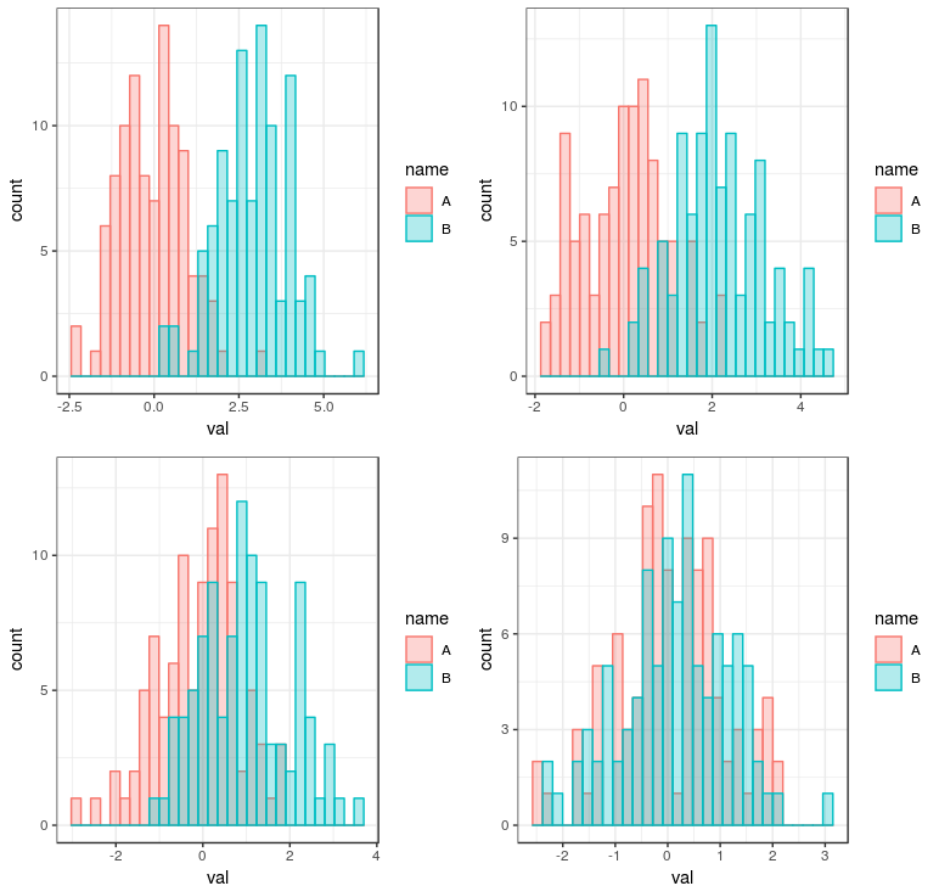
A			B		
TP=63	FP=28	91	TP=77	FP=77	154
FN=37	TN=72	109	FN=23	TN=23	46
100	100	200	100	100	200
TPR = 0.63			TPR = 0.77		
FPR = 0.28			FPR = 0.77		
PPV = 0.69			PPV = 0.50		
F1 = 0.66			F1 = 0.61		
ACC = 0.68			ACC = 0.50		

C			C′		
TP=24	FP=88	112	TP=76	FP=12	88
FN=76	TN=12	88	FN=24	TN=88	112
100	100	200	100	100	200
TPR = 0.24			TPR = 0.76		
FPR = 0.88			FPR = 0.12		
PPV = 0.21			PPV = 0.86		
F1 = 0.23			F1 = 0.81		
ACC = 0.18			ACC = 0.82		



Krzywa ROC

ROC można również wykorzystać do oceny pojedynczego klasyfikatora w formie krzywej ROC.



Rozpatrzmy przypadek LDA:

- dla klasyfikatora otrzymujemy wartości prawdopodobieństw a posteriori $p(1|x)$ oraz $p(2|x)$ przynależności do klas
- zwykle zakładamy, że jeśli $p(1|x) > T = 1/2$, to obserwacja zostaje zaliczona do klasy 1;
- możemy jednak manewrować parametrem T, zmieniając jego wartość od 0 do 1,
- otrzymamy wtedy zestaw macierzy pomyłek

Krzywa ROC

Wartości *FPR* oraz *TPR* otrzymane z tych macierzy utworzą krzywą. Pole pod krzywą (*AUC - area under curve*) świadczy o jakości klasyfikatora.