

Raport z projektu „Kółko i krzyżyk”

Celem projektu jest sprawdzenie skuteczności różnych klasyfikatorów za pomocą gry „Kółko i krzyżyk”. Rozważamy wszystkie możliwe zakończenia gry. Celem klasyfikatorów będzie stwierdzenie, czy przy danym ustawieniu na planszy gracz grający „x” wygrał czy nie. Wprowadzamy następujące założenia:

- Gracz grający „x” zaczyna
- Gra się kończy w momencie, gdy jeden z graczy wygra -> mogą być puste pola
- Są dwie klasy – gracz grający „x” wygrywa, bądź nie
- 958 możliwych gier
- b – puste pole
- positive – gracz grający „x” wygrał
- negative – gracz grający „x” nie wygrał

Przykładowy fragment danych:

617	b,b,o,o,o,x,x,x,x,positive
618	b,b,o,o,b,b,x,x,x,positive
619	b,b,o,b,o,b,x,x,x,positive
620	b,b,o,b,b,o,x,x,x,positive
621	b,b,b,x,x,x,o,o,b,positive
622	b,b,b,x,x,x,o,b,o,positive
623	b,b,b,x,x,x,b,o,o,positive
624	b,b,b,o,o,b,x,x,x,positive
625	b,b,b,o,b,o,x,x,x,positive
626	b,b,b,b,o,o,x,x,x,positive
→ 627	x,x,o,x,x,o,o,b,o,negative
628	x,x,o,x,x,o,b,o,o,negative
629	x,x,o,x,x,b,o,o,o,negative
630	x,x,o,x,o,x,o,o,b,negative
631	x,x,o,x,o,x,o,b,o,negative
632	x,x,o,x,o,o,o,x,b,negative
633	x,x,o,x,o,o,o,b,x,negative
634	x,x,o,x,o,o,b,x,o,negative

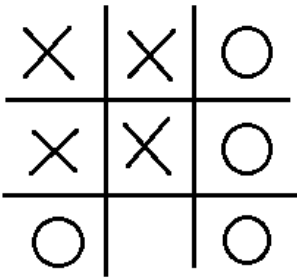


Fig. 1

Wiersz oznaczony strzałką odpowiada ułożeniu z prawej strony.

Używano następujących klasyfikatorów:

- ❖ Liniowa analiza dyskryminacji (LDA)
 - dla PU = 75%, 50% oraz 25% wszystkich danych
- ❖ Kwadratowa analiza dyskryminacji (QDA)
 - dla PU = 75%, 50% oraz 25% wszystkich danych
- ❖ Naiwny Bayes (GNB)
 - dla PU = 75%, 50% oraz 25% wszystkich danych
- ❖ Drzewo klasyfikujące
 - dla PU = 75%, 50% oraz 25% wszystkich danych
- ❖ Metoda najbliższych sąsiadów (K-NN)
 - dla k = 1, 5, 10, przy PU = 50% wszystkich danych
- ❖ Maszyny wektorów podpierających (SVM)
 - dla c = 1, 5, 10, przy PU = 50% wszystkich danych

Zatem w sumie sprawdzano 18 różnych metod.

Wyniki dla testu „accuracy” z Pythona na Próbie Testowej, uśrednione po N=10000 razy:

Name	Score
SVM c=10	0.960798
SVM c=5	0.954485
Drzewo 75% PU	0.933659
Drzewo 50% PU	0.911226
SVM c=1	0.871417
Drzewo 25% PU	0.857047
K-NN k=5	0.832306
K-NN k=1	0.75476
QDA 50% PU	0.725651
QDA 75% PU	0.724595
LDA 75% PU	0.717252
GNB 75% PU	0.716005
QDA 25% PU	0.714721
K-NN k=10	0.713108
LDA 50% PU	0.712204
GNB 50% PU	0.704668
LDA 25% PU	0.699382
GNB 25% PU	0.689756

Fig. 2

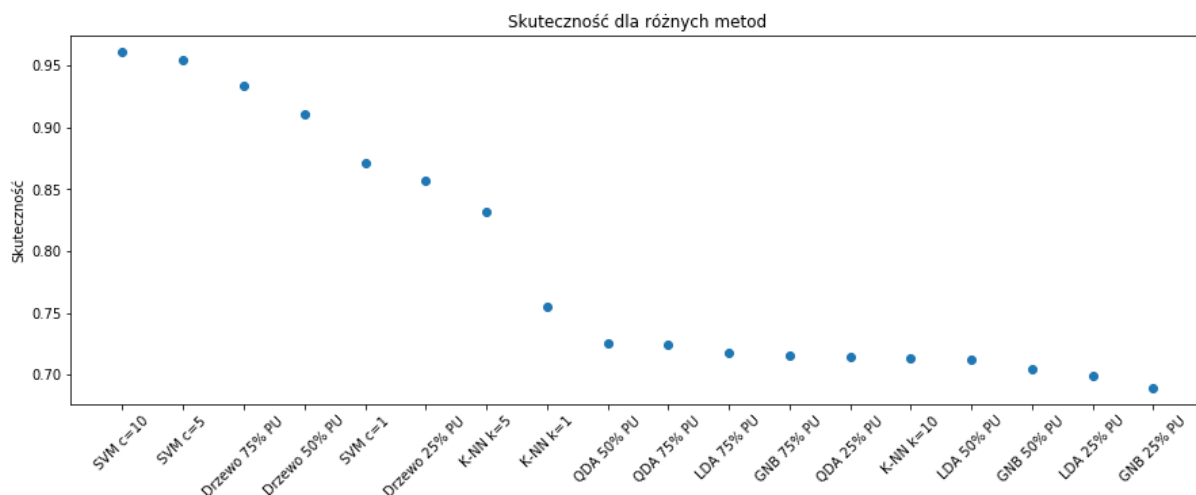


Fig. 3

Można wywnioskować, że najlepszą dokładność ma SVM (w szczególności dla $c=10$ oraz $c=5$). Im większy parametr kosztu, tym lepszy wynik.

Drzewa uczące są również bardzo dobre, im więcej danych prześlemy na PU, tym lepiej.

Na trzecim miejscu jest K-NN, ale tutaj już nie ma zależności między liczbą sąsiadów, a skutecznością.

LDA, QDA i naiwny Bayes są bardzo zbliżone i nie ma zbyt dużej różnicy między wielkością PU i PT.

Wyniki dla testu „accuracy” z Pythona na Próbie Testowej wraz z krosvalidacją dla Próby Uczącej, uśrednione po $N=1000$ razy:

Name	Accuracy	Crossvalidation
SVM c=10	0.96091	0.941483
SVM c=5	0.95441	0.935841
Drzewo 75% PU	0.934117	0.922059
Drzewo 50% PU	0.912669	0.898988
SVM c=1	0.871674	0.836192
Drzewo 25% PU	0.857517	0.832559
K-NN k=5	0.83299	0.78469
K-NN k=1	0.755918	0.756278
QDA 50% PU	0.725623	0.72476
QDA 75% PU	0.725372	0.725926
LDA 75% PU	0.717351	0.716266
GNB 75% PU	0.715172	0.710316
QDA 25% PU	0.714927	0.710704
K-NN k=10	0.713366	0.69821
LDA 50% PU	0.712611	0.710585
GNB 50% PU	0.703847	0.700866
LDA 25% PU	0.699209	0.698641
GNB 25% PU	0.689533	0.688596

Fig. 4

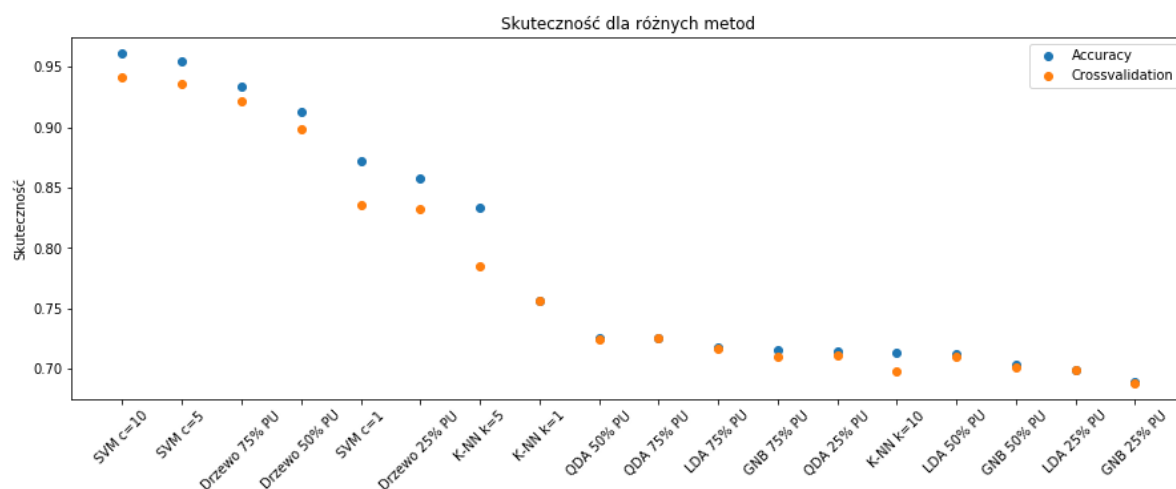


Fig. 5

Dla klasyfikatorów z lepszą skutecznością, krosvalidacja jest nieco mniejsza, co teoretycznie wskazuje na to, że mimo wszystko są one niedouczzone. Dla reszty krosvalidacja pokrywa się ze skutecznością.

Podsumowanie:

Generalnie dla problemu 2-klasowego skuteczność jest stosunkowo wysoka. Liczności klas są różne, więc może mieć to wpływ na dokładność dopasowania. Dla tego typu problemów najlepszymi klasyfikatorami są SVM oraz drzewa uczące.