# Indukcyjna Analiza Danych
# Laboratorium 4
# Algorytm:
# K-najbliższy sąsiadów
# K-nearest neighbours

Piotr Błoński

21 maja 2020

Prowadzący: Dr inż. Paweł Myszkowski

## 1 Cel ćwiczenia

Zapoznanie się z metodą klasyfikacji k-najbliższych sąsiadów (knn, k-nearest neighbours) na przy samodzielnej implementacji przy użyciu języka Python.

## 2 Plan badań

a. Implementacja algorytmu Knn

b. Implementacja 3 sposobów głosowania :

   (a) większościowe równoprawne

   (b) ważone odległością

   (c) ważone różnicą odległości najbliższego i najdalszego sąsiada danej klasy.

c. Implementacja 2 miar długość : Euklidesowa i Manhatan.

d. Użycie kroswalidacji stratyfikowanej.

e. Użycie standaryzacji

# 3 Knn - wstęp teoretyczny

Jest to jeden z podstawowych algorytmów nie parametrycznych używany w klasyfikacji i regresji. Należy do kategorii uczenia 'instance-based'. Jego działanie jest dość proste :

a. Przed przekazaniem danych należy je znormalizować / standaryzować

b. Dla zadanej nowej obserwacji obliczamy odległość do znanych już obserwacji za pomocą zadanej funkcji obliczania dystansu. Np. Odległość euklidesowa, Manhattan i inne.

c. Następnie wybieramy K najbliżej leżących obserwacji - dalej nazywamy je 'sąsiadami'.

d. Korzystając z którejś metody głosowania wybieramy do jakieś klasy będzie przynależeć Nowa obserwacja.

Jeśli chodzi o K czyli liczbę sąsiadów to powinniśmy celować w liczbę parzystą aby zmniejszyć ryzyko wystąpienia 'remisów' podczas głosowania. Podczas wystąpienia remisu jedną z metod radzenia sobie z nimi jest użycie wagi (np. dystans do najbliższego) lub zmienienie K. Generalnie im większe K tym knn będzie bardziej odporny na szumy i outlierów kosztem złożoności obliczeniowej. Musimy także pamiętać aby K nie było zbyt duże bo może zdarzyć się sytuacja że K będzie większe od ilości obserwacji danej klasy przez co obserwacja może nie zostać sklasyfikowana poprawnie.

## 3.1 Lazy learning

Tego terminu używa się do algorytmów uczenia maszynowego które nie tworzą swoistego 'modelu' czyli wyuczonych zestawów parametrów. Podczas 'lazy learnigu' algorytm tylko 'zapamiętuje' dane które są mu podane, nie dokonuje aproksymacji żadnej funkcji. Dopiero w podczas używania modelu model wykonuje większość obliczeń, co skutkuje tym że są o wiele wolniejsze od zwykłych modeli podczas używania, jednakże proces nauki to w większości po prostu podanie danych z odroczeniem obliczeń na później.

## 3.2 Koszt

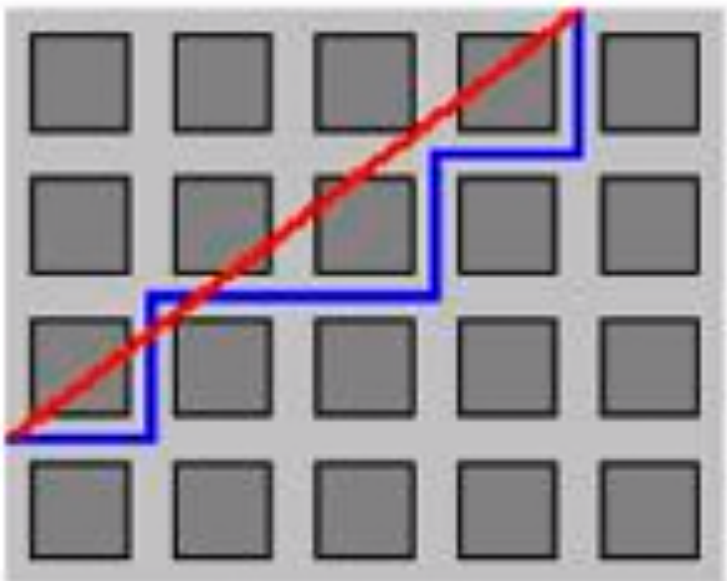
Niestety, ponieważ Knn musi przy każdej predykcji obliczyć odległość każdej obserwacji od nowej obserwacji, jest dość kosztowny obliczeniowo. Im więcej danych tym dłużej zajmuje wykonanie tej operacji, ale także powinna zwiększyć się jego skuteczność.

# 4 Funkcje obliczania odległości

Note: Jako że dane mamy numeryczne posłużymy się Euclidian i Manhattan distance. Gdybyśmy mieli dane kategoryczne moglibyśmy zastosować np Hamming'a. Obydwie te funkcje należą do kategorii Minkowski distance (p-norm). Gdzie dla Euclidian p=2 a dla manhatann p=1.

$$d\left(p,q\right) = \sqrt[p]{\sum_{i=1}^{n}\left(q_i - p_i\right)^p} \tag{1}$$

Niestety funkcje te są dość wrażliwe na outlierów.



Rysunek 1: Porównanie dystansów Manhatann(niebieski) i Euclidian(czerwony)

Miara dystansu powinna spełniać warunki:

a. Powinna być dodatnia

b. Równa 0 tylko i wyłącznie wtedy gdy x=y

c. Symetryczna

d. $d(x,z) <= d(x,y) + d(y,z)$ Traingle inequality

$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n}|p_i - q_i|$

# 5 Standaryzacja

Jest to proces przed uczeniem maszynowym w którym 'wyrównujemy dane' tak aby ich rozkład miał średnią wartość = 0 i odchylenie standardowe równe 1.
$z = \frac{x-\mu}{\sigma}$ gdzie z to nowa standaryzowana wartość, x to zmienna niestandaryzowana, $\mu$ to średnia z populacji a $\sigma$ to odchylenie standardowe populacji.

# 6 Knn - pytania pomocnicze

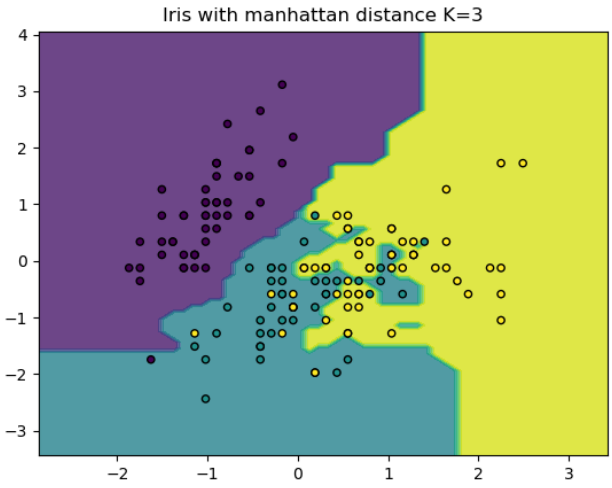
a. Co jest modelem (klasyfikacji) w algorytmie k-nn? - modelem jest algorytm który po dostaniu danych umie wykonać obliczenia które odpowiedzą na klasyfikacje/regresje.

b. Jak miary odległości wpływają wpływają na skuteczność modelu? -

c. Dlaczego zwykle nie stosuje się parzystych k? - aby uniknąć ryzyka remisów w głosowaniu

# 7 Wykonane badania

W ramach zajęć laboratoryjnych zostały zmierzone metryki: precision, recall, fscore dla kombinacji 2 różnych funkcji dystansów, 3 metod głosowania, K=[3,5,$\sqrt{N}$],Foldów kroswalidacji = [3,5,10]. Gdzie N jest ilością obserwacji w zbiorze.
Zbiory użyte w badaniach to: Wine, Glass i Seed. A także Iris dla testów czy algorytm w ogóle działa prawidłowo.
Użyto także Standaryzacji z biblioteki Sklearn. Podczas kroswalidacji z każdego foldu zbierane są metryki i następnie uśredniane dla każdej kombinacji.
Każde badanie zostało zapisane jako osobny plik, następnie przygotowanym skryptem są wczytywane, sortowane i filtrowane do datasecie. Poniżej znajdują się top10 wyników dla każdego datasetu. (W sumie 4 tabele po 10 rekordów. W przypadku chęci spojrzenia na pozostałe wyniki zapraszam do dołączonych plików)




Rysunek 2: Przykładowe klastry dla zbioru Iris

# 8 Wnioski

Pragnę zaznaczyć że w załączniku do sprawozdania w postaci kodu i wersji pdf znajdują się także wyniki wyższych foldów które nie zostały wzięte pod uwagę podczas pisania sprawozdania ze względu na bardzo długi czas ewaluacji i zagorzeniu czytelności sprawozdania.

| Parametry i zbiór | precision | recall | fscore |
|---|---|---|---|
| Iris K:sqrt(N) Folds:8 większościowe euclidean | 0.962302 | 0.960317 | 0.960220 |
| Iris K:5 Folds:3 większościowe euclidean | 0.961057 | 0.959967 | 0.960136 |
| Iris K:5 Folds:5 ważone najdalszym dystansem manhatan | 0.962424 | 0.960000 | 0.959933 |
| Iris K:sqrt(N) Folds:3 większościowe manhattan | 0.965465 | 0.959559 | 0.959899 |
| Iris K:5 Folds:5 większościowe euclidean | 0.963434 | 0.960000 | 0.959832 |
| Iris K:sqrt(N) Folds:5 większościowe euclidean | 0.964646 | 0.960000 | 0.959798 |
| Iris K:5 Folds:8 ważone najdalszym dystansem euclidean | 0.967262 | 0.958333 | 0.959355 |
| Iris K:3 Folds:8 ważone najdalszym dystansem euclidean | 0.956349 | 0.955357 | 0.954545 |
| Iris K:3 Folds:8 większościowe euclidean | 0.956349 | 0.955357 | 0.954545 |
| Iris K:5 Folds:8 większościowe manhattan | 0.956349 | 0.955357 | 0.954545 |

| Parametry i zbiór | precision | recall | fscore |
|---|---|---|---|
| Glass K:3 Folds:5 ważone dystansem manhattan | 0.680137 | 0.641786 | 0.636845 |
| Glass K:5 Folds:5 ważone dystansem manhattan | 0.679428 | 0.641964 | 0.630183 |
| Glass K:sqrt(N) Folds:8 ważone dystansem manhattan | 0.696665 | 0.679977 | 0.620221 |
| Glass K:5 Folds:5 ważone dystansem euclidean | 0.636474 | 0.651865 | 0.617526 |
| Glass K:sqrt(N) Folds:5 ważone dystansem manhattan | 0.662155 | 0.667321 | 0.612685 |
| Glass K:3 Folds:8 ważone dystansem manhattan | 0.714749 | 0.637731 | 0.610956 |
| Glass K:sqrt(N) Folds:3 ważone dystansem manhattan | 0.618418 | 0.655764 | 0.606612 |
| Glass K:3 Folds:5 ważone dystansem euclidean | 0.625369 | 0.623512 | 0.604242 |
| Glass K:sqrt(N) Folds:3 ważone dystansem euclidean | 0.624226 | 0.622972 | 0.602911 |
| Glass K:3 Folds:3 ważone dystansem manhattan | 0.614848 | 0.626664 | 0.598928 |

| Parametry i zbiór | precision | recall | fscore |
|---|---|---|---|
| Wine K:3 Folds:8 większościowe euclidean | 0.425458 | 0.284495 | 0.274193 |
| Wine K:3 Folds:8 ważone najdalszym dystansem euclidean | 0.486075 | 0.283422 | 0.271899 |
| Wine K:3 Folds:5 większościowe euclidean | 0.372848 | 0.272893 | 0.268607 |
| Wine K:5 Folds:5 ważone najdalszym dystansem manhatan | 0.555918 | 0.265374 | 0.268461 |
| Wine K:5 Folds:5 ważone najdalszym dystansem euclidean | 0.455939 | 0.263961 | 0.266614 |
| Wine K:3 Folds:5 ważone najdalszym dystansem euclidean | 0.439865 | 0.274481 | 0.265806 |
| Wine K:5 Folds:8 ważone najdalszym dystansem euclidean | 0.494576 | 0.261169 | 0.263784 |
| Wine K:5 Folds:8 ważone najdalszym dystansem manhatan | 0.508829 | 0.262497 | 0.262673 |
| Wine K:5 Folds:5 większościowe euclidean | 0.604225 | 0.262083 | 0.261101 |
| Wine K:5 Folds:8 większościowe manhattan | 0.536864 | 0.262490 | 0.259116 |

| Parametry i zbiór | precision | recall | fscore |
|---|---|---|---|
| seeds K:5 Folds:3 większościowe euclidean | 0.931249 | 0.923913 | 0.923804 |
| seeds K:sqrt(N) Folds:5 większościowe manhattan | 0.924792 | 0.919048 | 0.920322 |
| seeds K:sqrt(N) Folds:5 większościowe euclidean | 0.925201 | 0.919048 | 0.919924 |
| seeds K:sqrt(N) Folds:3 większościowe euclidean | 0.924683 | 0.918881 | 0.919291 |
| seeds K:3 Folds:5 ważone najdalszym dystansem euclidean | 0.925110 | 0.919048 | 0.919230 |
| seeds K:3 Folds:5 większościowe euclidean | 0.925110 | 0.919048 | 0.919230 |
| seeds K:sqrt(N) Folds:8 większościowe manhattan | 0.926389 | 0.918981 | 0.918012 |
| seeds K:sqrt(N) Folds:8 większościowe euclidean | 0.924151 | 0.918981 | 0.917949 |
| seeds K:5 Folds:8 większościowe euclidean | 0.923958 | 0.918403 | 0.917711 |
| seeds K:3 Folds:8 większościowe euclidean | 0.929335 | 0.919560 | 0.915790 |

## 8.1 Wyniki z poprzednich laboratorium

| Parametry i zbiór | Fscore |
|---|---|
| Iris | 0.946 |
| Glass | 0.736 |
| Wine | 0.83 |
| Seed | 0.77 |

Wyniki dla naiwnego bayesa z kroswaldacją 5

Fold

a. Iris - najlepsze wyniki Iris osiąga dla głosowania większościowego, i dystansem euclidean. Różnice pomiędzy wielkością K i foldów pomiędzy dwoma najlepszymi sięga dziesięcio-tysięczynych więc nie ma dużego znaczenia. Jest lepszy niż powyższy naiwny bayes.

b. Glass - najlepsze wyniki Glass osiąga dla głosowania ważonego dystansem manhatann. Niestety wyniki dla 10 najlepszych są w przedziale (0.59,063) co nie jest fenomenalnym wynikiem. Może to być spowodowane ilością wymiaró w Glass. Niestety jest gorszy od naiwnego bayesa.

c. Wine - najlepsze wyniki Wine osiąga dla głosowanie większościowego i dystansu euclidean, jednakże 6/10 najlepszych wyników to głosowanie ważone najdalszym dystansem. Mierzenie dystansu jako euclidean 7/10 najlepszych wyników. Niestety wyniki są tragiczne i nie osiągają nawet 0.3. Naiwny bayes jest zdecydowanie lepszy z wynikiem 0.83. Topowe wyniki są osiągane dla 8 i 5 foldów, gdzie dla 8 są najlepsze.

d. Seed - najlepsze wyniki osiągnął dla głosowanie większościowego i miary euclidean. Połowa najlepszych wyników to K równe sqrt z ilości danych. Seed preferuje niższe foldy (3 i 5). Osiąga aż o 0.2 lepsze wyniki niż naiwny bayes.