Komputerowe wspomaganie diagnozowania białaczek u dzieci z wykorzystaniem algorytmów minimalno-odległościowych

Łukasz Odwrot(*218283*), Jarosław Ciołek-Żelechowski(*218386*)

***Streszczenie*—Celem pracy jest stworzenie klasyfikatora minimalno-odległościowego mogącego wspomóc diagnozowanie białaczki u dzieci. Dane do opisywanego zadania zostały dostarczone wcześniej przez prowadzącego zajęcia i zostały opisane w osobnym rozdziale. Wyniki działania klasyfikatora zostały opisane na podstawie przykładowego eksperymentu.**

I. Wstęp

Praca składa się z omówienia zastosowanego algorytmu, definicji problemu, opisania cech i przedstawieniu ich rankingu, samego badania/eksperymentu wraz z opisem wpływu zmiennych na jakość wyników, prezentacji i omówienia rezultatów.

Wykorzystanie systemów uczących się w diagnostyce medycznej nie jest zjawiskiem nowym. Dane medyczne od dawna są przechowywane w celu ponownego wykorzystania w przypadku wystąpienia podobnych objawów wśród różnych pacjentów. W tym celu dane przechodzą obróbkę, zostają sklasyfikowane i ujednolicone. Tak zbudowany system, nawet w obrębie pojedynczego szpitala jest w stanie wytworzyć ilość danych pozwalającą na zbudowanie i *wytrenowanie* dowolnego klasyfikatora [**1**].

II. Omówienie algorytmu

W pracy posłużyliśmy się dwoma najpopularniejszymi algorytmami minimalno-odległościowymi:

- NM(ang. *nearest mean*) - najbliższej średniej,
- KNN(ang. *k nearest neighbors*) - k najbliższych sąsiadów

Ideą tych algorytmów jest klasyfikacji danego osobnika do odpowiadającej mu klasy na podstawie minimalnych odległości do pewnych elementów swojego otoczenia.

Najbliższej centroidy - dla każdej klasy na podstawie wszystkich obiektów, wyliczana jest średnia centroida. Obiekt przypisywane jest na podstawie minimalnej odległości do tej centroidy.

K najbliższych sąsiadów - ze znanych obiektów wybieramy k najbliższych klasyfikowanemu obiektowi. Nowy obiekt zostaje przypisany do klasy, w której znajduje się najwięcej z k reprezentantów.

A. Miary odległości

Jako że mamy do czynienia z algorytmami opartymi na mierze długości - dobrze by było sobie tą miarę zdefiniować. My w naszym badaniu korzystamy z następujących metryk:

- **Euklidesowa**, czyli zwykła odległość odcinka łączącego dwa punkty opisana wzorem:

$$d(A,B) = \sqrt{\sum_{i=1}^{n}((x_{iA} - x_{iB})^2)}$$

- **Manhattan**, czyli suma wartości bezwzględnych różnic współrzędnych dwóch punktów(można to sobie wyobrazić jako ruch taksówki w mieście o układzie ulic w szachownice). Wzór:

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

, gdzie $\mathbf{p}$ i $\mathbf{q}$ są wektorami.

III. Definicja problemu rozpoznawania

Nr cechy	Nazwa cechy	Możliwe wartości
1	Temperatura	1, 2
2	Anemia	1, 2, 3
3	Stopień krwawienia	1, 2
4	Miejsce krwawienia	1, 2, 3, 4, 5, 6, 7, 8
5	Bóle kości	1, 2
6	Wrażliwość mostka	1, 2
7	Powiększenie węzłów chłonnych	1, 2
8	Powiększenie wątroby	1, 2
9	Centralny układ nerwowy	1, 2
10	Powiększenie jąder	1, 2
11	Uszkodzone serce	1, 2
12	Gałka oczna	1, 2
13	Poziom WBC	1, 2, 3
14	Poziom RBC	1, 2, 3
15	Płytk krwi	1, 2
16	Niedojrzałe komórki	1, 2
17	Pobudzenie szpiku	1, 2, 3
18	Główne komórki	1, 2, 3
19	Poziom limfocytów	1, 2, 3
20	Reakcja	1, 2

Tablica I: Opis cech

Z dostarczonych Nam danych wszystkie cechy mają charakter dyskretny, przy czym te o numerach: **1**, **3**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **15**, **16**, **20** mają właściwości **binarne**(przyjmują wartość 1 lub 2), a te o numerach: **2**, **4**, **13**, **14**, **17**, **18**,

19 są cechami **wielowartościowymi**. Wszystkie cechy zostały opisane w tabeli I

Wymienione cechy pozwalają na klasyfikację danego osobnika do jednej z 20 klas przedstawionych w tabeli II

Nr	Jednostk chorobowa
1	L1 - type
2	L2 - type
3	L3 - type
4	Undifferentation
5	Differentation in part
6	Granuocylosisi
7	Granua mononucleaw
8	Mononucleacyble
9	Redikaukemia
10	Subatue grandblacyta
11	Granulacytarna
12	Lymphocytia
13	Granue mononclea
14	Mononuclea
15	Lymphosarcoma leukemia
16	Pamacea Leukemia
17	Multicapilary be laukemia
18	Acicople granulotyne leukemia
19	Basaphi granulocyte leukemia
20	Macronuclacycle teuekema

Tablica II: Jednostki chorobowe

A. Normalizacja

To proces wstępnej obróbki danych, gdzie wszystkie cechy są sprowadzone do tego samego zakresu od 0 do 1. Czyli w naszej sytuacji dla np. *Miejsca krwawienia*, które normalnie opisywane jest przy pomocy liczby ze zbioru: [1, 2 , 3 , 4, 5 , 6, 7 , 8], teraz będzie opisywane zbiorem: [0.125, 0.25, 0.373, 0.5, 0.625, 0.75, 0.875, 1.0]

Opisane dalej eksperymenty przeprowadzaliśmy dla danych znormalizowanych i nie. Jednym z elementów eksperymentu jest porównanie i określenie wpływu tego zabiegu.

IV. Ranking cech

Selekcja cech dla danych ma 3 najważniejsze zalety:

- **Redukcja przeładowania** - mniej niepotrzebnych/nieznaczących danych oznacza podejmowanie decyzji w oparciu o znaczące dane, a nie o *szum*,
- **Zwiększenie celności** - mniej mylących danych oznacza lepszą celność/dokładność,
- **Redukcja czasu trenowania** - mniej danych = szybszy algorytm.

Jaki jednak algorytm rankingowy dla cech zastosować?

- **Testy Statystyczne** - sprawdzenie zależności poszczególnej cechy a wynikowej klasy w oparciu o test statystyczny np. test chi-kwadrat,
- **Rekurencyjna eliminacja cech** - sprawdzenie zależności poprzez budowanie modelu, przeprowadzeniu statystyk, zbudowaniu ponownie modelu z mniejszą ilością cech i porównanie statystyk. Im lepsze statystyki tym lepszy zestaw cech. Metoda czasochłonna.,
- **Analiza głównych składowych** - interpretacja zbioru jako chmury N-punktów(obserwacji) w przestrzeni K-wymiarowej(zmiennych). Zadaniem algorytmy jest takie obrócenie układu współrzędnych by maksymalizować wariancję pierwszej współrzędnej, a następnie kolejnych.

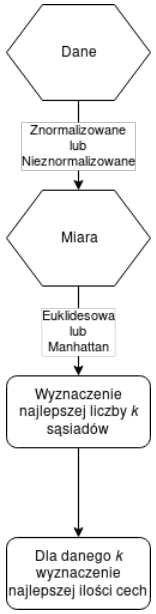
Dla naszych danych przeprowadziliśmy ranking cech przy pomocy testu statystycznego chi-kwadrat. W tabeli III przedstawiliśmy wynik rankingu cech dla danych znormalizowanych i nie.

Dane znormalizowane	Dane nieznormalizowane
Stan pobudzenia szpiku	Miejsce krwawienia
Anemia	Stan pobudzenia szpiku
Gałka oczna	Poziom limfocytów
Obniżenie poziomu RBC	Anemia
Główne komórki szpiku	Obniżenie poziomu RBC
Poziom limfocytów	Główne komórki szpiku
Liczba płytek krwi	Gałka oczna
Wrażliwość mostka	Liczba płytek krwi
Powiększenie jąder	Wrażliwość mostka
Reakcja	Powiększenie jąder
Poziom WBC	Reakcja
Stopień krwawienia	Poziom WBC
Powiększenie węzłów chłonnych	Uszkodzenie w sercu
Uszkodzenie w sercu	Ból kości
Ból kości	Powiększenie węzłów chłonnych
Centralny układ nerwowy	Niedojrzałe komórki
Miejsce krwawienia	Centralny układ nerwowy
Niedojrzałe komórki	Powiększenie wątroby
Powiększenie wątroby	Stopień krwawienia
Temperatura	Temperatura

Tablica III: Ranking cech

V. Opis eksperymentów

Eksperymenty przeprowadziliśmy w następujący sposób:



Rysunek 1: Diagram Eksperymentu

Jak widać mamy do czynienia z dwoma zmiennymi, tj. typ danych i użyta miara. Dla każdej z tych kombinacji przeprowadzamy dwuetapowy eksperyment.

A. Implementacja

W celu przeprowadzenia badania badań i wykonania potrzebnych pomiarów skorzystaliśmy z języka **Python** i obecnych dla niego bibliotek: **sklearn**, **pandas**, **matplotlib**, **seaborn**.

Wybór Pythona był spowodowany łatwością i szybkością implementacji, zarówno od strony dostępnych rozwiązań(biblioteki z których korzystaliśmy miały zaimplementowane wszystkie potrzebne nam algorytmy), jak i naszych umiejętności(znaliśmy oboje najlepiej właśnie Pythona).

Kod całej aplikacji dostępny jest na repozytorium [**2**]

B. Walidacja krzyżowa

Trenowanie i testowanie klasyfikatorów opisanych w tej pracy odbyło się z wykorzystaniem 5x2cv, czyli 5-krotnej walidacji krzyżowej. Oznacza to że 5 razy losowo podzieliliśmy zbiór danych na dane uczące i testujące, wykonaliśmy badania i zamieniliśmy zbiory uczące i testujące miejscami.

C. Miary jakości

W celu określenia jakości otrzymanego wyniku, korzystaliśmy z szeregu statystyk:

- **accuracy** - procent poprawnych klasyfikacji. Stosunek ilości poprawnych predykcji dla danej klasyfikacji, opisana wzorem:

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

  , gdzie y to wartość prawdziwa, a $\hat{y}$ to wartość naszej predykcji,
- **precision** - umiejętność klasyfikatora do niepoprawnej klasyfikacji próbek negatywnych jako pozytywnych, najprościej opisać wzorem:

$$\frac{tp}{(tp + fp)}$$

  , gdzie tp to ilość prawdziwie pozytywnych predykcji(poprawnie sklasyfikowane *prawdy*), a fp to ilość fałszywie pozytywnych(niepoprawnie sklasyfikowanych *prawd*),
- **recall** - umiejętność klasyfikatora do poprawnej klasyfikacji, opisana wzorem:

$$\frac{tp}{(tp + fn)}$$

  , gdzie tp to ilość prawdziwie pozytywnych predykcji(poprawnie sklasyfikowane *prawdy*), a fn to ilość fałszywie negatywnych(niepoprawnie sklasyfikowanego *fałszu*),
- **fscore** - miara bardziej skomplikowana, będąca średnią ważoną dwóch powyższych,

$$F1 = \frac{2 * (precision * recall)}{(precision + recall)}$$

- **confusionMatrix** - tablica błędów. Graficzna reprezentacja wszystkich wykonanych predykcji w postaci macierzy stanu faktycznego i wyliczonego przez nasz klasyfikator. Dla idealnej klasyfikacji zaznaczona jest tylko przekątna takiej macierzy.

W celu rozróżnieniu dwóch wyników w trakcie eksperymentu porównywaliśmy jego **fscore** i wybieraliśmy ten z większym wynikiem.

VI. Przebieg eksperymentu

A. Wyznaczenie najlepszej liczby k sąsiadów

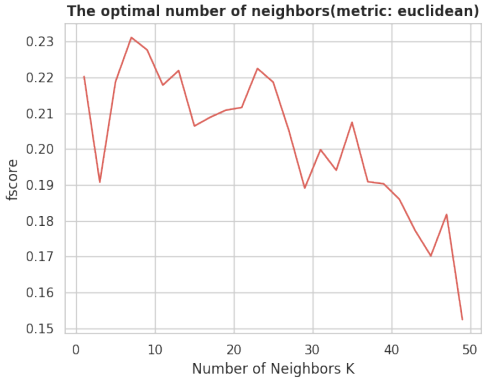
Ten krok eksperymentu przeprowadziliśmy tylko dla algorytmu k-najbliższych sąsiadów. Dla algorytmu Najbliższej centroidy nie było sensu go przeprowadzać - brak parametru k.

Stworzyliśmy listę k-sąsiadów. Lista składała się z nieparzystych liczb z zakresu 1 -50. Nieparzystych, by etap podejmowania decyzji przez klasyfikator nie był uzależniony od losowości(sytuacji gdy dany element ma taką samą ilość różnych sąsiadów wokół). Następnie dla każdej liczby z tego zakresu przeprowadziliśmy klasyfikację i wybraliśmy jako najlepszą tą, której wynik **fscore** był największy.

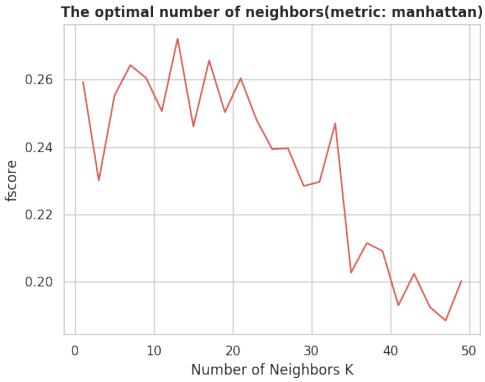
Wyniki naszych obliczeń przedstawione są na rysunkach 2, 3, 4 i 5, oraz przedstawione w tabeli IV(Legenda typów: **1 - Znormalizowane, Euklides**; **2 - Znormalizowane, Manhattan**; **3 - Nieznormalizowane, Euklides**; **4 - Nieznormalizowane, Manhattan**)

Typ	Najlepsze k	Accuracy	Precision	Recall	Fscore
1	7	0.269	0.331	0.269	0.261
2	13	0.273	0.302	0.273	0.257
3	1	0.251	0.286	0.251	0.249
4	9	0.260	0.310	0.260	0.247

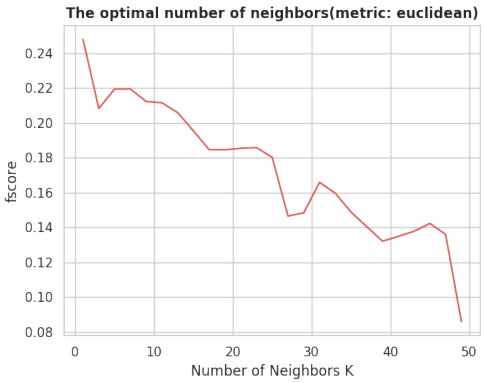
Tablica IV: Wyniki wyznaczania najlepszej liczby k sąsiadów



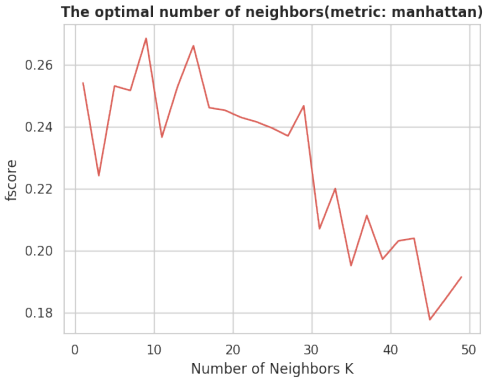
Rysunek 2: Dane: znormalizowane, miara: euklidesowa



Rysunek 3: Dane: znormalizowane, miara: manhattan



Rysunek 4: Dane: nieznormalizowane, miara: euklidesowa



Rysunek 5: Dane: nieznormalizowane, miara: manhattan

B. *Dla danego k wyznaczenie najlepszej ilości cech*

Ten krok eksperymentu przeprowadziliśmy zarówno dla algorytmu k-najbliższych sąsiadów jak i dla algorytmu najbliższej centroidy. Przy czym w tym drugim wypadku nie parametryzowaliśmy wielkości k, jako że jej nie ma.

Ideą było wykorzystanie rankingu cech z tabeli III i wyliczenie dla nich statystyk w oparciu o wcześniej wyznaczone k

Wyniki przedstawione są na rysunkach 6, 7, 8, 9 dla danych znormalizowanych, i na rysunkach 10, 11, 12, 13 dla danych nieznormalizowanych. Oraz zebrane w tabeli V i VI(Legenda typów: **1 - Znormalizowane, Euklides**; **2 - Znormalizowane, Manhattan**; **3 - Nieznormalizowane, Euklides**; **4 - Nieznormalizowane, Manhattan**)

Typ	Liczba cech	Najlepsze k	Acc.	Prec.	Rec.	F-scr
1	10	7	0.260	0.308	0.260	0.246
2	19	13	0.279	0.313	0.279	0.264
3	20	1	0.249	0.260	0.249	0.242
4	18	9	0.264	0.308	0.264	0.253

Tablica V: Algorytm k - najbliższych sąsiadów - optymalna liczba cech

Typ	Liczba cech	Acc.	Prec.	Rec.	F-scr
1	19	0.291	0.302	0.291	0.282
2	12	0.252	0.284	0.252	0.241
3	20	0.215	0.272	0.215	0.213
4	17	0.281	0.304	0.281	0.275

Tablica VI: Algorytm Najbliższych centroidów - optymalna liczba cech

VII. Omówienie wyników

Zastosowane algorytmy klasyfikacji dla Naszych danych nie przynoszą zadowalających wyników. Uzyskujemy celność na poziomie 30%

A. *Wyznaczenie K*

Jak widać na tabeli IV najlepsze wyniki uzyskuje się dla relatywnie małej ilości sąsiadów. Co więcej rysunki 2, 3, 4, 5 pokazują tendencję malejącą wraz ze wzrostem liczby sąsiadów, co pokrywa się to wynikami opisanymi w literaturze [**3**]

B. *Wyznaczenie Liczby cech*

Jak widać na tabelach V, VI najlepsze wyniki uzyskujemy dla dużej liczby cech, patrząc jednak na wykresy 6, 7, 8, 9 widać ewidentnie tendencję że dodawanie nowych cech dość szybko zwiększa jakoś predykcji, jednak równie szybko dochodzimy do momentu w którym uzyskujemy wyniki sub optymalne i dodawanie nowych cech zmienia jakość tylko nieznacznie.

C. *Wpływ normalizacji*

Już w momencie gdzie wybieramy optymalna wartość k dane znormalizowane uzyskują lepsze wyniki od danych nieznormalizowanych, patrz tabela IV. Po ustaleniu optymalnej ilości cech tendencja ta nadal się utrzymuje(tabela V, VI).

Warto jednak zaznaczyć że różnice nie są duże. Wynika to z tego że i tak większość naszych cech ma charakter binarnych, więc jest niejako znormalizowana.

D. Wpływ użytej miary

Zbierając wyniki z tabel IV V, VI nie można jednoznacznie wybrać lepszej miary. Dla wybierania optymalnego k lepiej spisuję się miara euklidesowa zarówna dla danych znormalizowanych jak i nie. Dla optymalizacji cech, dla obu typów danych, najlepsze wyniki uzyskuje miara manhattan. Optymalizacja cech dla algorytmu najbliższej centroidy dla danych znormalizowanych wykazuje lepsze wyniki dla miary euklidesowej, podczas gdy dla danych nieznormalizowanych ten sam algorytm lepiej wykonuje predykcję dla miary manhattan.

Warto zaznaczyć że wyniki dla obu miar są zbliżone.

E. Macierz Błędów

Dla wszystkich eksperymentów, stworzyliśmy macierze błędów widoczne na rysunkach 14 - 21. Jak widać na każdej z nich można dostrzec zarys mocniej zaznaczonej przekątnej.

VIII. Wnioski

Najlepsze wyniki dla algorytmu k - najbliższych sąsiadów uzyskaliśmy dla danych znormalizowanych, z wykorzystaniem miary manhattan i zostały zaprezentowane w tablicy VII.

Liczba cech	Najlepsze k	Acc.	Prec.	Rec.	F-scr
19	13	0.279	0.313	0.279	0.264

Tablica VII: najlepszy wynik KNN

Dla algorytmu najbliższej centroidy, najlepsze wyniki dały dane znormalizowane z miarę euklidesową i zostały zaprezentowane w tablicy VIII.

Liczba cech	Acc.	Prec.	Rec.	F-scr
19	0.291	0.302	0.291	0.282

Tablica VIII: Najlepszy wynik NM

Zgodnie z wynikami omówionymi w poprzednim rozdziale, stwierdzamy że:

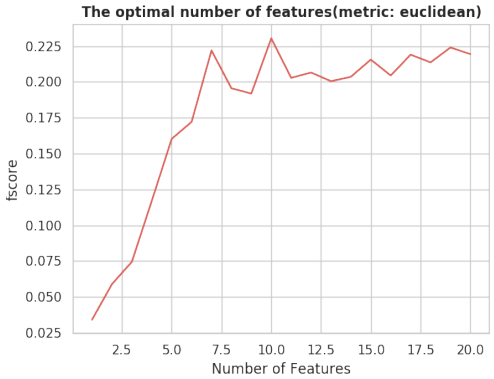
- Normalizacja pozytywnie wpływa na wyniki predykcji,
- Użyta mira nie wpłynęła znaczącą na wyniki,
- Klasyfikator lepiej sobie radzi dla niższych wartości parametru k,
- Ilość cech dość szybko dochodzi do momentu w którym uzyskiwane wyniki są sub optymalne, potem zależność ta się wypłaszcza.

Bibliografia

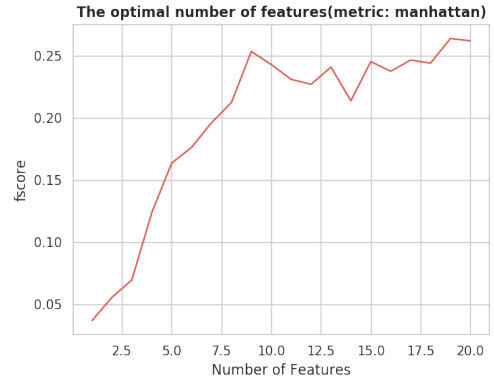
[1] J. Clin Pathol, 'Guide to Medical Informatics, the Internet and Telemedicine', Apr. 1998.

[2] Repozytorium, https://github.com/lOdwrot/MedycynaBialaczka

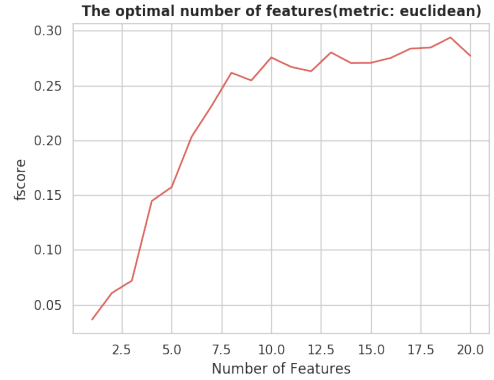
[3] Paweł Cichosz, 'Systemy uczące się', Wydawnictwo WNT, 2009.



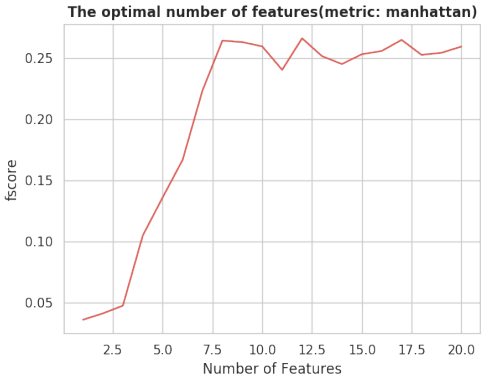
Rysunek 6: Dane: znormalizowane, miara: euklidesowa



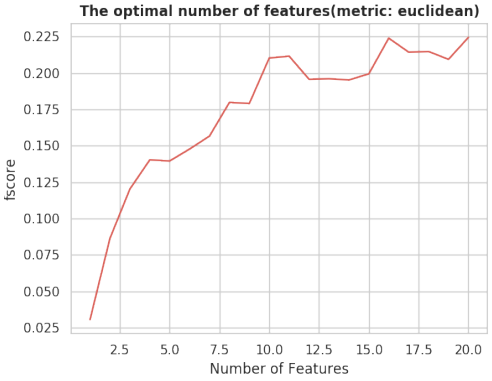
Rysunek 7: Dane: znormalizowane, miara: manhattan



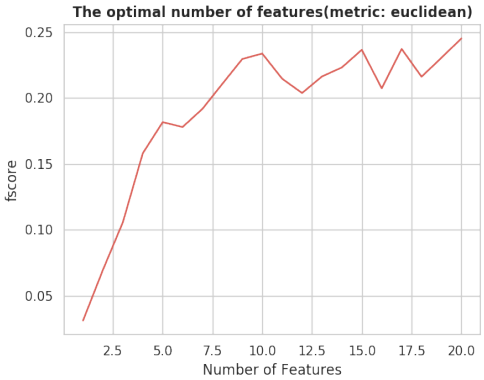
Rysunek 8: Dane: znormalizowane, miara: euklidesowa



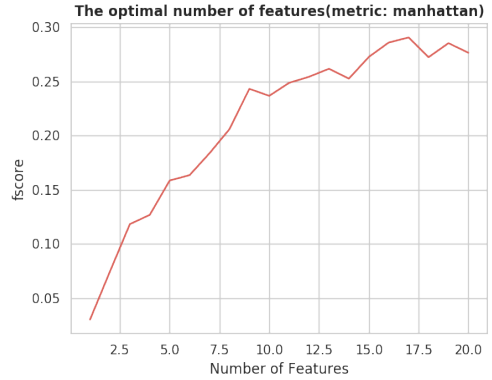
Rysunek 9: Dane: znormalizowane, miara: manhattan



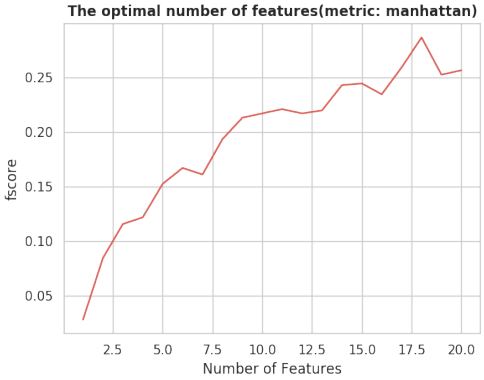
Rysunek 12: Dane: nieznormalizowane, miara: euklidesowa



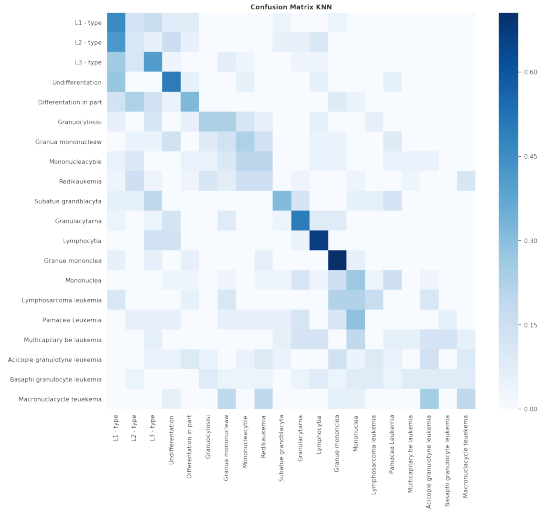
Rysunek 10: Dane: nieznormalizowane, miara: euklidesowa



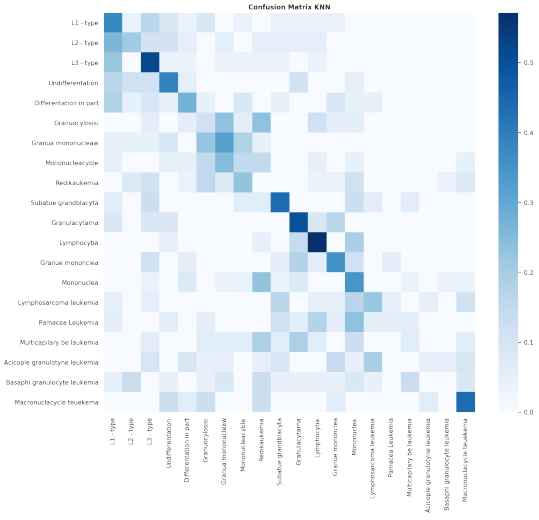
Rysunek 13: Dane: nieznormalizowane, miara: euklidesowa



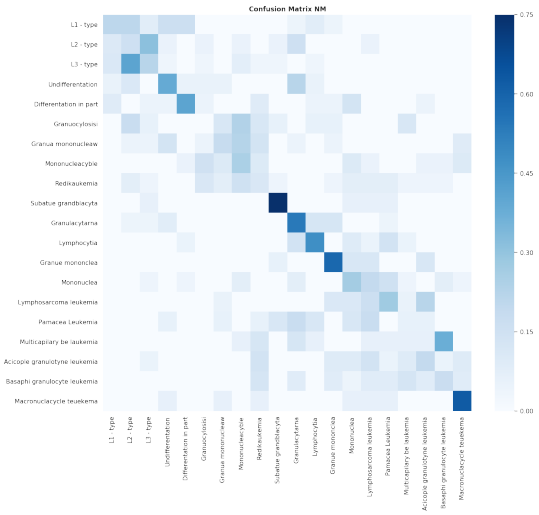
Rysunek 11: Dane: nieznormalizowane, miara: manhattan



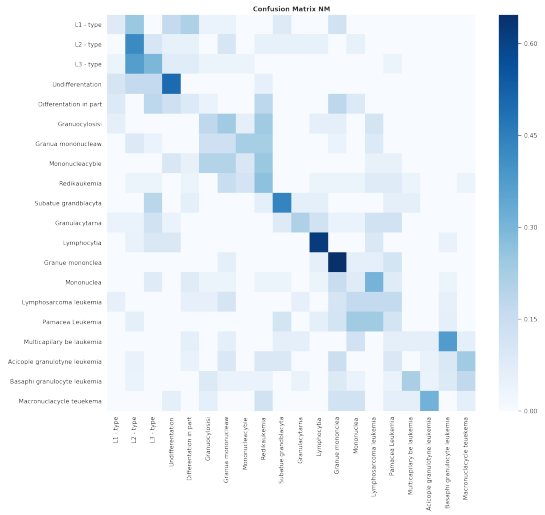
Rysunek 14: Dane: znormalizowane, miara: euklidesowa, algorytm: KNN



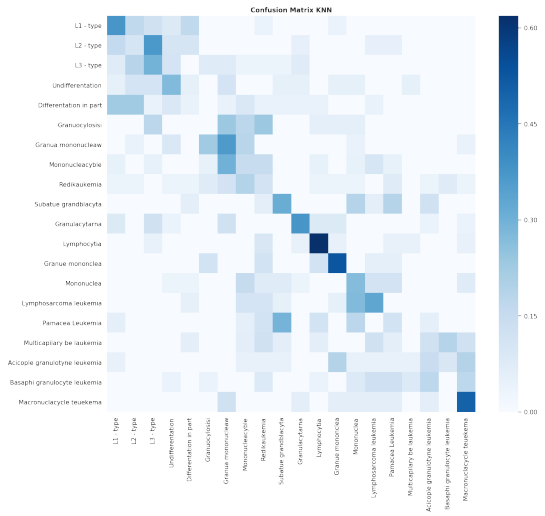
Rysunek 15: Dane: znormalizowane, miara: manhattan, algorytm: KNN



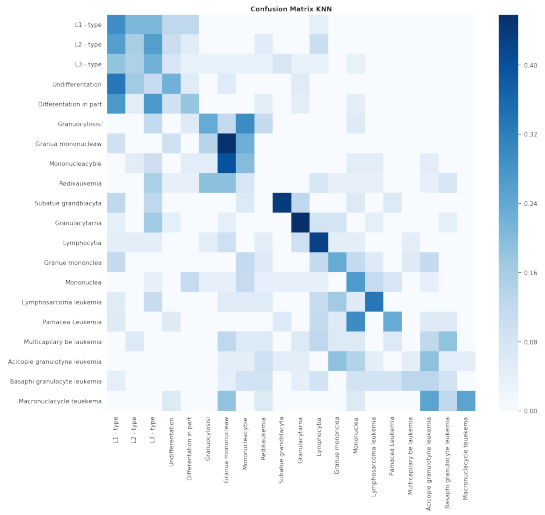
Rysunek 16: Dane: znormalizowane, miara: euklidesowa, algorytm: NM



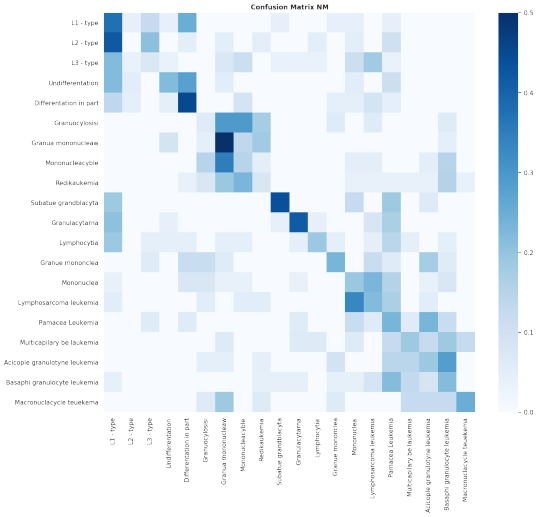
Rysunek 17: Dane: znormalizowane, miara: manhattan, algorytm: NM



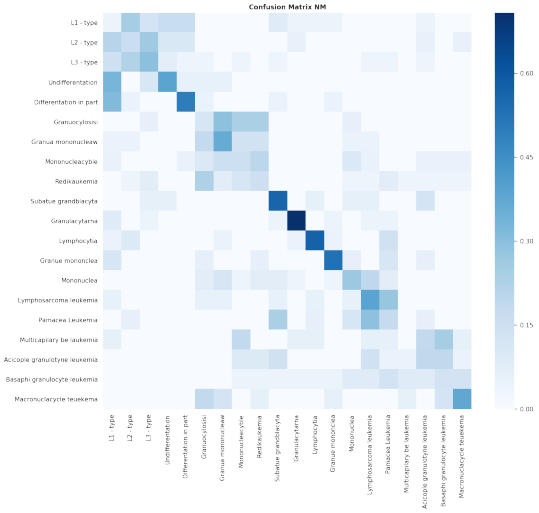
Rysunek 18: Dane: nieznormalizowane, miara: euklidesowa, algorytm: KNN



Rysunek 19: Dane: nieznormalizowane, miara: manhattan, algorytm: KNN



Rysunek 20: Dane: nieznormalizowane, miara: euklidesowa, algorytm: NM



Rysunek 21: Dane: nieznormalizowane, miara: manhattan, algorytm: NM