Politechnika Wrocławska

Wydział Elektroniki

Zastosowania Informatyki w Medycynie

Komputerowe wspomaganie diagnozowania zawałów z wykorzystaniem algorytmu KNN

Autorzy:

Bartosz Rodziewicz	226105
Kamil Dobrysiewicz	225961

Prowadzący:

Dr inż. Paweł Ksieniewicz

Spis treści

1 Założenia projektowe

Celem niniejszego projektu jest nabycie umiejętności zastosowania algorytmu klasyfikacji nadzorowanej (w przypadku tego projektu algorytmu KNN) w zadaniu diagnozowania zawałów. Wymaga to odpowiedniej selekcji cech. Dostępność danych rzeczywistych umożliwi w przyszłości eksperymentalną ocenę skuteczności algorytmu i sprawdzenie, w jaki sposób jakość klasyfikacji zależy od liczby atrybutów wykorzystanych do skonstruowania modelu.

Wyróżniono następujące etapy realizacji projektu:

1. Zapoznanie się z aglorytmem klasyfikacji, określonym w temacie projektu.
2. Zapoznanie się z materiałem empirycznym - analiza danych wejściowych, określenie liczby i znaczenia klas oraz dokonanie charakterystyki cech.
3. Opracowanie sposobu wyznaczania rankingu cech w wykorzystaniem rozwiązań dostępnych w bibliotece `scikit-learn`.
4. Zaplanowanie badań eksperymentalnych.
5. Implementacja algorytmu klasyfikacji.
6. Przeprowadzenie badań eksperymentalnych.
7. Analiza wyników i wyciągnięcie wniosków.
8. Przygotowanie kompletnej dokumentacji.

2 Charakterystyka analizowanego problemu

Do badań wykorzystane będą dane dostarczone przez prowadzącego, zawierające 901 obiektów. Podzielono je na pięć plików, reprezentujących dostępne w badaniach diagnozy:

- dusznicę bolesną,
- dusznicę odmienną (Prinzmetala),
- zawał mięśnia sercowego (pełnościenny),
- zawał mięśnia sercowego (podwsierdziowy),
- ból nie związany z sercem.

W każdym z zestawów danych znajdują się informacje o obiektach opisanych za pomocą 59 cech, oznaczających wyniki badań pojedyńczego pacjenta. Cechy podzielono na 8 grup, które opisują:

- dane o wieku i płci pacjenta,
- informacje o bólu, który wystąpił u chorego (lokalizacja, promieniowanie, charakter bólu, czas trwania ostatniego wystąpienia bólu),
- inne symptomy, które wystąpiły razem z bólem (nudności, pocenie się, odbijanie),
- historię wystąpień podobnego bólu (bóle związane z zawałem, dusznicą bolesną, powiązane z sercem),
- historię chorób pacjenta (występowanie zawałów w przeszłości, przewlekła niewydolnośc serca, nadciśnienie),
- informacje o obecnie zażywanych lekach (beta blokery, diuretyki, niesteroidowe leki przeciwzapalne),
- wyniki badania fizykalnego (ciśnienie krwi, tętno, sinica, szmery oddechowe),
- wyniki badania elektrokardiografem (EKG).

Większość cech ma charakter binarny, czyli posiada tylko dwie wartości (0 lub 1), np. płeć, czy pacjent zażywa beta blokery, czy chory ma nadciśnienie. Jest to najprostsza odmiana atrybutu kategorycznego. W zbiorze cech znaleźć można też kilka cech kategorycznych, które przyjmować mogą kilka wartości z grupy możliwych opcji, np. lokalizacja bólu, moment wystąpienia bólu, jego charakter, czy kierunek promieniowania. Wyróżnić można także cechy ciągłe, takie jak: wiek pacjenta, liczba godzin od rozpoczęcia bólu, ciśnienie skurczowe, tętno.

3 Opis zastosowanych algorytmów

3.1 Metryki odległości

W grupie algorytmów minimalno-odległościowych, do której należy algorytm k-NN istotną rolę odgrywa zastosowana metryka odległości, wg której mierzone są odległości pomiędzy badanymi punktami. Spośród metryk dostpęnych w bibliotece *scikit-learn* wybrano odległość Euklidesową oraz metrykę miejską (zwaną inaczej odległością Manhattan).

3.1.1 Odległość Euklidesowa

Odległość Euklidesowa stanowi jeden z najpopularniejszych sposobów obliczania odległości między obiektami w przestrzeni wielowymiarowej. Jej wartość obliczana jest za pomocą wzoru:

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

Odległość Euklidesowa jest więc równa długości odcinka, który łączy dwa dane punkty.

3.1.2 Metryka miejska

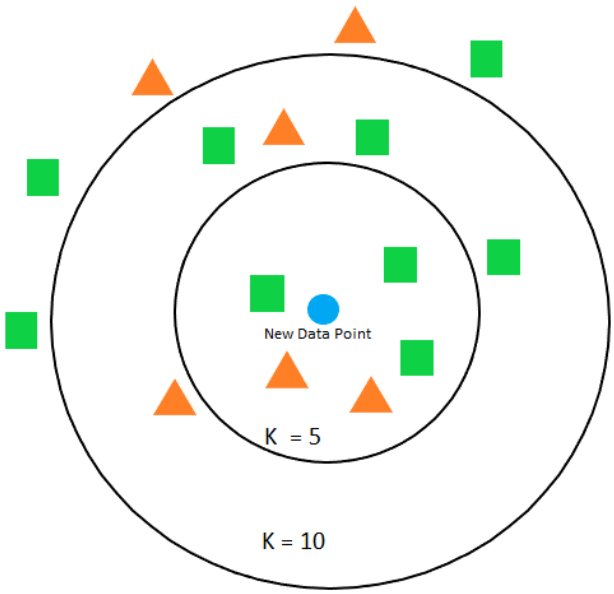
Metryka miejska to sposób obliczania odległości, gdzie możliwe jest poruszanie się tylko w dwóch prostopadłych do siebie kierunkach. Inna nazwa tej metryki to odległość Manhattan, ponieważ przypomina ona poruszanie się po ulicach Manhattanu. Jej wartość obliczana jest ze wzoru:

$$d(A,B) = \sum_{i=1}^{n}|x_{Ai} - x_{Bi}|$$

3.2 Algorytm K-NN

Algrytm K-NN (k nearest neighbours - k najbliższych sąsiadów) jest jednym z nadzorowanych algorytmów uczenia maszynowego. Jego działanie opiera się na bardzo prostej idei przewidywania nieznanych wartości poprzez ich dopasowanie do najbardziej podobnych już znanych wartości. W przypadku poszukiwania najbardziej podobnego rozwiązania uzyskujemy algorytm 1NN. Zazwyczaj warto jednak wziąć pod uwagę kilka lub kilkanaście podobnych rozwiązań i wybrać rozwiązanie najbardziej popularne w tym zbiorze. Podobieństwo jest w tym przypadku obliczane na podstawie metryki odległości, jaka została przez nas wybrana, na przykład wcześniej omówionej odległości Euklidesowej, czy metryki miejskiej.

Na rysunku 1 zaprezentowany został prosty przykład problemu klasyfikacji przy pomocy algorytmu KNN. W przypadku, kiedy wartość k wynosi 5, niebieski okrąg reprezentujący niesklasyfikowany obiekt zostanie przyporządkowany do zbioru zielonych kwadratów (3 kwadraty w pobliżu, zaś tylko 2 trójkąty). Do podobnej klasyfikacji dojdzie w przypadku, gdy wartość k wyniesie 10. Wtedy obiekt testowy również zostanie uznany za zielony kwadrat, których w pobliżu niebieskiego okręgu znajdzie się 6.



Rysunek 1: Przykład klasyfikacji przy pomocy algorytmu KNN.

4 Stworzenie rankingu cech

W niniejszym projekcie ranking cech wyznaczany był dla każdego zbioru testowego z osobna, ponieważ nie jest możliwe wyznaczenie jednego optymalnego rankingu cech dla całego zbioru danych.
Do stworzenia każdego z rankingów wykorzystano klasę **SelectKBest** z biblioteki scikit-learn. Algorytm filtrowania cech, jaki wykorzystano to χ^2-distribution. Algorytm chi-squared został wybrany spośród dostępnych w bibliotece scikit-learn, ponieważ radzi sobie lepiej niż inne algorytmy z tej biblioteki ze zmiennymi nieciągłymi.
Metoda generująca ranking cech miała jednocześnie za zadanie usunięcie cech uznanych za nieistotne ze zbioru uczącego i testowego. Jako dane wejściowe przyjmuje zbiór danych uczących (osobno informacje o cechach i diagnozy) oraz liczbę najlepszych cech, które mają zostać wzięte pod uwagę w procesie klasyfikacji. Przebieg tej metody wygląda następująco:

1. Stworzenie obiektu klasy **SelectKBest** z ustawieniem algorytmu filtrowania cech jako *chi2* oraz liczby najlepszych cech, jakie mają zostać zwrócone (na podstawie parametru wejściowego metody).
2. Wykonanie metody klasyfikującej dla zbioru uczącego.
3. Pobranie identyfikatorów kolumn, zawierających najlepsze cechy.
4. Eliminacja cech nieistotnych ze zbioru uczącego i testowego. Modyfikacja zbioru testowego została dokonana w celu zachowania spójności z zawartością zbioru uczącego.

Metoda zwraca zredukowany zbiór uczący i testowy, na którym potem pracuje klasyfikator.

W tabeli 1 przedstawiono zaś ranking cech, opracowany na początkowym etapie realizacji projektu, który został wyznaczony w oparciu o wszystkie dostępne obiekty.

Tabela 1: 25 najlepszych cech wg rankingu wyznaczonego dla wszystkich obiektów.

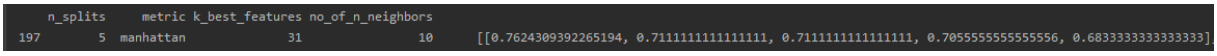
Numer cechy	Nazwa cechy	Wartość χ^2
35	Systolic blood pressure	1980.23
6	Number of hours since onset	978.58
2	Pain location	340.52
53	New ST segment depression	223.47
49	New Q wave	200.26
55	New T wave inversion	193.40
51	New ST segment elevation	188.22
54	Any ST segment depression	177.00
57	New intraventricular conduction defect	159.89
56	Any T wave inversion	151.67
38	Respiration rate	120.30
43	Diastolic murmur	117.16
58	Any intraventricular conduction defect	117.10
21	Prior angina prectoris	116.64
3	Chest pain radiation	114.72
37	Heart rate	109.56
50	Any Q wave	108.82
45	S3 gallop	105.09
17	Prior pain related to heart	101.63
18	Prior pain due to MI	89.17
46	S4 gallop	87.11
19	Prior pain due to angina prectoris	85.87
23	Congestive heart failure	85.27
42	Systolic murmur	84.50
25	Hiatal hernia	83.61

5 Implementacja klasyfikatora

Samo wywołanie klasyfikatora nie okazało się bardzo wymagającym zadaniem, dzięki wykorzystaniu biblioteki scikit-learn, która oferuje bardzo szeroki wachlarz gotowych funkcjonalności w tym zakresie. Konieczne było jednak odpowiednie przygotowanie danych wejściowych dla klasyfikatora. Następnie przygotowano metodę, która przyjmuje na wejściu zbiór cech dostępnych obiektów badawczych, diagnozy dla tych zespołów cech, oraz zmienne istotne z punktu widzenia procesu badawczego, takie jak liczba sąsiadów, liczba najlepszych cech, jakie mają być brane pod uwagę w procesie klasyfikacji, czy wybrana metryka obliczania odległości między sąsiadami. W algorytmie zastosowano strategię dwóch powtórzeń pięciokrotnej walidacji krzyżowej. Podziału zbioru wejściowego na 5 części dokonano z wykorzystaniem klasy **StratifiedKFold**. Umożliwiła ona podział zbioru wejściowego na pięć podzbiorów, przy czym stosunek ilości obiektów zakwalifikowanych do poszczególnych klas jest dokładnie taki sam, jak dla całego zbioru przed podziałem. Przebieg metody wyznaczającej skuteczność klasyfikatora wygląda następująco:

1. Stworzenie obiektu klasy **StratifiedKFold** ze wskazaniem liczby grup, na jakie ma zostać podzielony zbiór danych wejściowych oraz ustawieniem zmiennej, która umożliwia jednakowe wyznaczenie grup dla wywołania głównej metody z innymi parametrami algorytmu.
2. Dla każdego zestawu indeksów danych uczących i testowych zwróconego przez metodę *split* klasy **StratifiedKFold** realizowane są kolejno następujące operacje:
   (a) Stworzenie dedykowanych zbiorów danych uczących i testowych (osobno zbiory cech i dedykowane im klasy) na podstawie ustalonych indeksów.
   (b) Generacja zbioru cech uczących i testowych na podstawie rankingu cech wyznaczanego indywidualnie dla każdego zadania klasyfikacji.
   (c) Stworzenie instancji klasyfikatora KNN z zadaną liczbą sąsiadów i wybraną metryką obliczania odległości.
   (d) Dopasowanie modelu na podstawie danych uczących.
   (e) Obliczenie skuteczności klasyfikacji danych testowych.

Algorytm zwraca macierz złożoną z wyników działania klasyfikatora o liczbie kolumn równej liczbie warstw, na jakie został podzielony zbiór wejściowy.
Przykładowy wynik uruchomienia algorytmu przedstawiono na rysunku 2:



	n_splits	metric	k_best_features	no_of_n_neighbors	
197	5	manhattan	31	10	[[0.7624309392265194, 0.7111111111111111, 0.7111111111111111, 0.7055555555555556, 0.6833333333333333]

Rysunek 2: Przykładowy wynik działania klasyfikatora.

6 Środowisko programistyczne

Środowisko programistyczne stanowił edytor Atom uruchomiony w systemie Windows. Algorytm był testowany z poziomu systemu Ubuntu zainstalowanego przez Windows Subsystem for Linux w systemie Windows. Uruchomienie algorytmu powiodło się również z poziomu wiersza polecenia w Windows 10. W projekcie wykorzystano język Python w wersji 3 oraz biblioteki scikit-learn, pandas i numpy. Badania wykononane zostały na komputerze stacjonarnym wyposażonym w procesor Intel Core i5 czwartej generacji oraz 8GB pamięci operacyjnej.

7 Opis badań eksperementalnych

Przygotowany klasyfikator poddany został badaniom skuteczności działania. Zweryfikowano poprawność klasyfikacji dla następujących parametrów:

- liczba sąsiadów - 1, 5, 10.
- metoda obliczania odległości między sąsiadami - odległość Euklidesowa i metryka miejska.
- liczba najlepszych cech w rankingu - od 1 do 35.

Badania przeprowadzono dla liczby najlepszych cech z przedziału od 1 do 35, ponieważ przy liczbie 29 cech udało się uzyskać najlepsze rezultaty, potem wyniki były już gorsze. Dla każdej kombinacji parametrów wejściowych uzyskano uśrednione wyniki względem 5 powtórzeń metody 2-krotnej walidacji krzyżowej.

8 Wyniki badań

W wyniku przeprowadzonych badań skuteczności klasyfikatora KNN otrzymano 210 rezultatów dla różnych kombinacji parametrów omówionych w poprzedniej sekcji. Każdy z wyników został uśredniony względem 2 powtórzeń 5-krotnej walidacji krzyżowej. W tabeli 2 przedstawiono ranking 25 najlepszych rezultatów, sporządzony na podstawie uśrednionych wyników działania klasyfikatora, zaokrąglonych do dwóch miejsc po przecinku.

Tabela 2: 25 najlepszych wyników działania klasyfikatora KNN.

Identyfikator testu	Metryka odległości	Liczba cech	Liczba sąsiadów	Skuteczność algorytmu (%)
197	Manhattan	31	10	72,36
203	Manhattan	33	10	72,25
188	Manhattan	28	10	72,2
205	Manhattan	34	5	72,19
206	Manhattan	34	10	72,14
209	Manhattan	35	10	72,09
202	Manhattan	33	5	72,08
190	Manhattan	29	5	71,75
191	Manhattan	29	10	71,7
193	Manhattan	30	5	71,64
194	Manhattan	30	10	71,64
208	Manhattan	35	5	71,53
196	Manhattan	31	5	71,47
199	Manhattan	32	5	71,42
200	Manhattan	32	10	71,31
178	Manhattan	25	5	71,25
173	Manhattan	23	10	71,25
176	Manhattan	24	10	71,2
175	Manhattan	24	5	71,14
170	Manhattan	22	10	70,87
181	Manhattan	26	5	70,87
179	Manhattan	25	10	70,86
187	Manhattan	28	5	70,7
182	Manhattan	26	10	70,7
172	Manhattan	23	5	70,64

Przedstawione rezultaty pozwalają na wyciągnięcie pewnych wniosków, jednak istnieje ryzyko, że niektóre z nich zostały zawyżone lub zaniżone poprzez wylosowanie sprzyjającego zbioru uczącego i testowego. W celu zwiększenia wiarygodności wyników postanowiono dokonać analizy statystycznej na zasadzie testów parowych ze skorygowanym testem t-Studenta dla powtórzonej walidacji krzyżowej.

8.1 Skorygowany test t-Studenta dla powtórzonej walidacji krzyżowej

W celu zwiększenia wiarygodności uzyskanych wyników dla każdej możliwej pary dwóch macierzy z wynikami działania algorytmu obliczona została statystyka na podstawie wzoru:

$$t = \frac{\overline{S}}{\sqrt{(\frac{1}{J*k} + \frac{N_{ts}}{N_{tr}}) * \sum_{i=1}^{k} \sum_{j=1}^{J} \frac{(S^{(i,j)} - \overline{S})^2}{k*J-1}}} \sim t_{k*J-1},$$

gdzie J stanowi liczbę powtórzeń (=2), k - liczbę warstw (=5), S - macierz różnicy wyników porównywanych testów, $\overline{S}$ - średnią wartość elementu macierzy S, $S^{(i,j)}$ - pojedynczy wynik dla i-tej części (foldu) i J-tego powtórzenia. Wartość $\frac{N_{ts}}{N_{tr}}$ została uproszczona na potrzeby ninejszego projektu i wynosi $\frac{1}{k}$, czyli 0,2.
Obliczone wartości testu t-Studenta porównane zostały z wartością krytyczną z tablicy rozkładu t-Studenta. Wartość krytyczna wyniosła 2,2622 (poziom prawdopodobieństwa - 5%, liczba stopni swobody - k * J - 1 = 9). Jeżeli wartość testu była większa niż wartość krytyczna to pierwszy wynik z pary uznawany był za statystycznie lepszy, jeżeli była mniejsza od ujemnej wartości krytycznej to drugi wynik uznawany był za lepszy, zaś jeżeli mieściła się w przedziale <ujemna wartość krytyczna; dodatnia wartość krytyczna> to wyniki uznawane były za jednakowo wartościowe.

8.2 Wyniki badań z uwzględnieniem rezultatów testów t-Studenta

Rezultatem obliczeń wartości testu t-Studenta była macierz 210x210, której komórki zawierają liczby ze zbioru {-1, 0, 1}, oznaczające kolejno kolejno: -1 - rezultat testu o numerze równym numerowi wiersza jest statystycznie gorszy od rezultatu testu o numerze równym numerowi kolumny, 0 - porównywane testy są statycznie jednakowo dobre, 1 - rezultat o numerze równym numerowi wiersza jest statystycznie lepszy od rezultatu o numerze równym numerowi kolumny. Fragment tej macierzy przedstawiono na rysunku 3:

	0	1	2	3	4	5	6	...	204	205	206	207	208	209	Sum
209	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	0.0	0.0	0.0	0.0	0.0	157.0
188	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	0.0	0.0	0.0	0.0	0.0	157.0
205	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	0.0	0.0	1.0	0.0	0.0	155.0
203	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	0.0	0.0	0.0	0.0	0.0	155.0
202	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	0.0	0.0	1.0	0.0	0.0	154.0
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
107	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-204.0
106	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-204.0
1	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-204.0
2	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-204.0
105	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-204.0

Rysunek 3: Fragment macierzy z interpretacją wyników testu t-Studenta.

Wyniki zawarte w otrzymanej macierzy zostały zsumowane i na ich podstawie ponownie został wyznaczony ranking skuteczości działania klasyfikatora KNN. 25 najlepszych wyników prezentuje tabela 3.

Tabela 3: 25 najlepszych wyników działania klasyfikatora KNN z uwzględnieniem wyników testu t-Studenta.

Identyfikator testu	Metryka odległości	Liczba cech	Liczba sąsiadów	Skuteczność algorytmu (%)
188	Manhattan	28	10	72,2
209	Manhattan	35	10	72,09
203	Manhattan	33	10	72,25
205	Manhattan	34	5	72,19
202	Manhattan	33	5	72,08
191	Manhattan	29	10	71,7
190	Manhattan	29	5	71,75
194	Manhattan	30	10	71,64
206	Manhattan	34	10	72,14
196	Manhattan	31	5	71,47
197	Manhattan	31	10	72,36
193	Manhattan	30	5	71,64
178	Manhattan	25	5	71,25
175	Manhattan	24	5	71,14
173	Manhattan	23	10	71,25
176	Manhattan	24	10	71,2
170	Manhattan	22	10	70,87
208	Manhattan	35	5	71,53
164	Manhattan	20	10	70,48
199	Manhattan	32	5	71,42
182	Manhattan	26	10	70,7
158	Manhattan	18	10	70,2
187	Manhattan	28	5	70,7
166	Manhattan	21	5	70,36
161	Manhattan	19	10	70,15

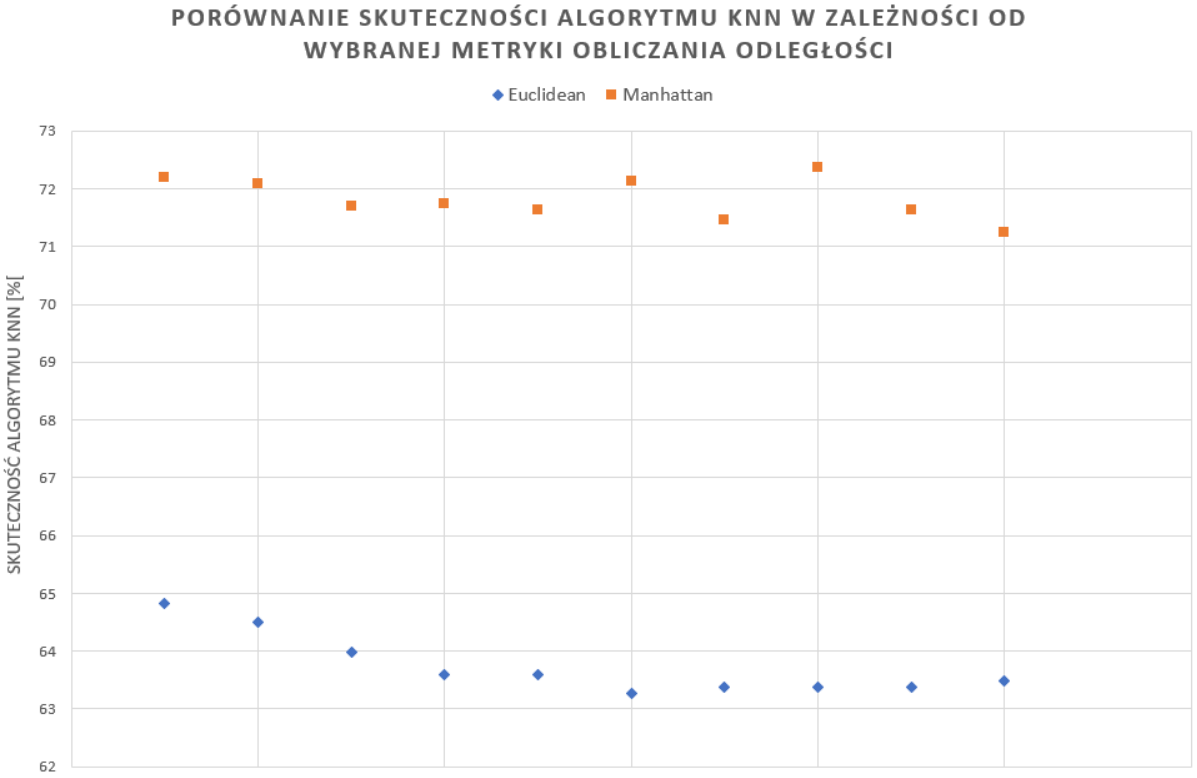
Dla najlepszego rezultatu wygenerowano przykładową macierz konfucji. Przedstawiono ją na rysunku 4:

	Wynik testu				
Prawdziwe dane	45	1	0	0	0
	7	13	3	5	1
	4	5	1	2	1
	0	1	0	51	1
	0	3	1	17	18

Rysunek 4: Macierz konfuzji dla statystycznie najlepszego zestawu parametrów wejściowych.

W macierzy konfuzji na zielono zaznaczono liczbę poprawnych przyporządkowań do danej klasy. Na jej podstawie można wyciągnąć wniosek, że klasyfikator najczęściej mylił się, klasyfikując podwsierdziowy zawał mięśnia sercowego jako zawał pełnościenny (33% przyporządkowań). Ponadto algorytm miał problem z diagnozowaniem dusznicy odmiennej. Było to prawdopodobnie spowodowane małą liczbą danych wejściowych dotyczących tej choroby.

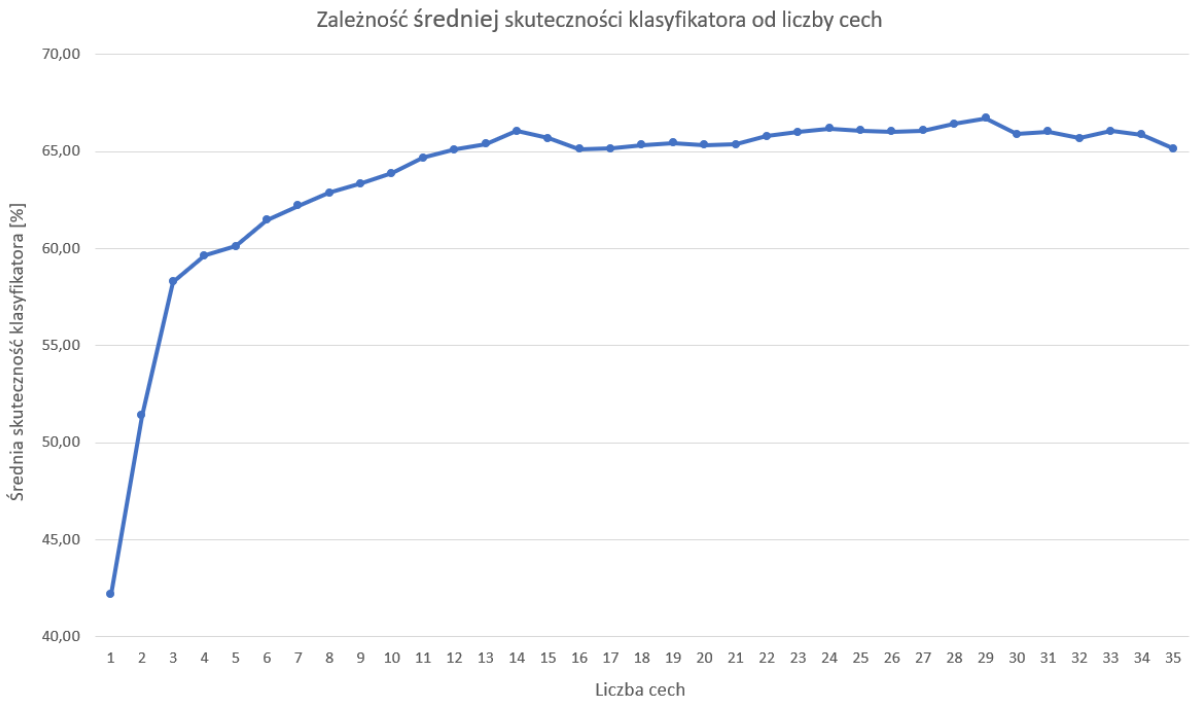
Na podstawie otrzymanych wyników można wyciągnąć wniosek, że dla badanego problemu lepszą metryką obliczania odległości między sąsiadami jest miara Manhattan. Wszystkie najlepsze wyniki uzyskane zostały właśnie dla tej metryki. Potwierdza to dodatkowo wykres najlepszych wyników dla obu miar odległości przedstawiony na rysunku 5:



Rysunek 5: Porównanie najlepszych wyników klasyfikatora KNN w zależności od metryki wyznaczania odległości.

Z rysunku 5 można ponadto odczytać, że metryka miejska pozwoliła na uzyskanie wyników o ok. 8% lepszych od metryki euklidesowej.

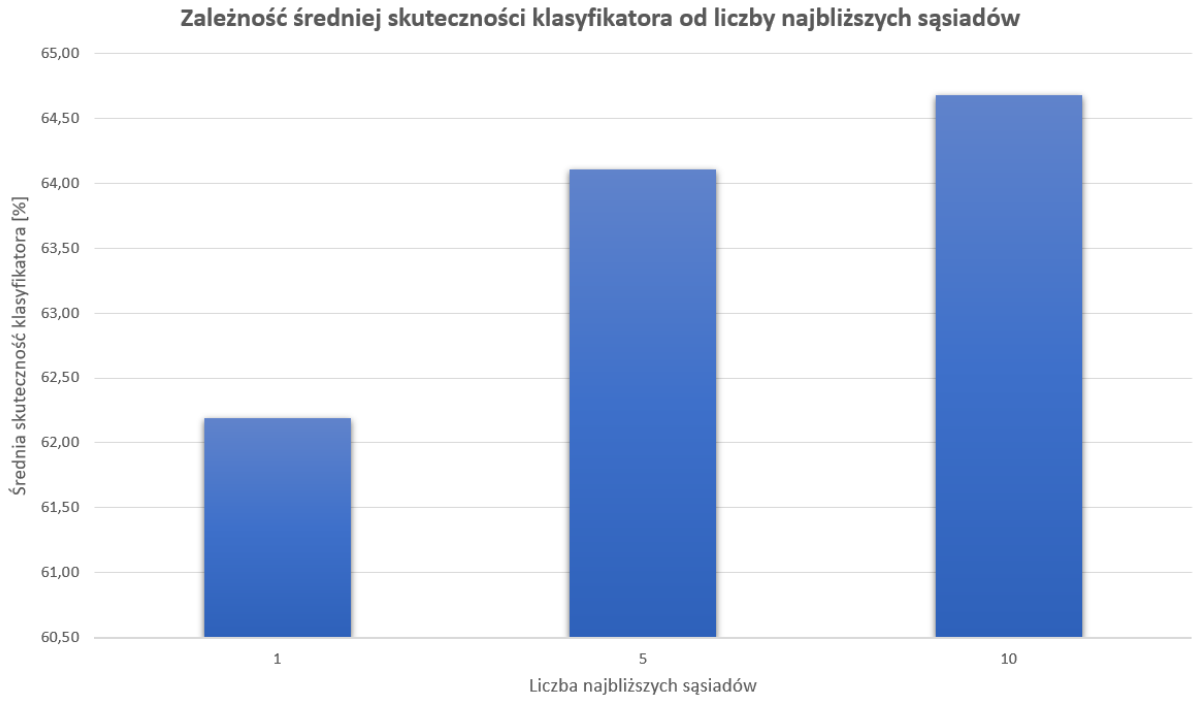
Kolejny wykres przedstawia zależność wyników od liczby cech branych pod uwagę w procesie klasyfikacji.



Rysunek 6: Wykres zależności wyników klasyfikacji od liczby cech branych pod uwagę.

Na podstawie wykresu przedstawionego na rysunku 6 można stwierdzić, że najwyższą skuteczność uzyskano dla 29 najlepszych cech z rankingu. Nieznacznie gorsze wyniki można uzyskać dla liczby cech z przedziału od 12 do 35, gdzies średnia skuteczność wynosi ok. 65%.

Wpływ liczby najbliższych sąsiadów na wyniki działania algorytmu przedstawiono na rysunku 7:



Rysunek 7: Wykres zależności wyników klasyfikacji od liczby najbliższych sąsiadów.

Na podstawie wykresu przedstawionego na rysunku 7 można wyciągnąć wniosek, że wzrost liczby najbliższych sąsiadów pozytywnie wpływa na wyniki klasyfikacji. Istnieje pewne prawdopodobieństwo, że dalsze zwiększanie wartości tego parametru mogłoby dodatkowo poprawić jej wyniki.

9 Wnioski

Na podstawie wykonanych badań oraz analizy ich wyników wyłoniony został zestaw cech, które pozwalają na uzyskanie najwyżej ze statystycznego punktu widzenia skuteczności klasyfikatora KNN dla problemu diagnozowania zawałów:

- Metryka obliczania odległości między sąsiadami - miara Manhattan.
- Liczba najistotniejszych cech z rankingu cech - 28.
- Liczba sąsiadów - 10.

Przedstawiony zestaw parametrów umożliwił uzyskanie średniej skuteczności na poziomie 72,2%. Najwyższa skuteczność uzyskana w procedurze testowej wyniosła jednak 72,36%, ale rezultat ten został uznany za mniej pewny na podstawie wyników testu t-Studenta. Algorytm KNN umożliwił rozwiązanie problemu diagnozowania zawałów u chorych z całkiem wysoką skutecznością.

Bardzo ważnym okazał się wybór metryki obliczania odległości między poszczególnymi sąsiadami. Metryka miejska pozwoliła uzyskać znacznie wyższą skuteczność klasyfikacji w porównaniu do metryki euklidesowej.

Wybór liczby najbliższych sąsiadów również miał zauważalny wpływ na jakość klasyfikacji, wzrost liczby sąsiadów skutkował wzrostem skuteczności działania klasyfikatora. W przyszłości warto byłoby przeprowadzić badania dla wyższych wartości tego parametru.

Liczba najlepszych cech z rankingu, branych pod uwagę w procesie klasyfikacji również wpływa na wyniki działania algorytmu, ale jest to wpływ zauważalnie niższy, niż w przypakdu pozostałych parametrów. Wybór liczby cech z zakresu od 12 do 35 pozwolił na uzyskanie podobnych wyników, jednak najlepsze wyniki zarejestrowano dla około 28 cech.

Literatura

[1] *https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf*, Selekcja cech - korelacja i współczynnik p.

[2] *https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b*, Selekcja cech w bibliotece `scikit-learn`.

[3] *http://home.agh.edu.pl/ horzyk/lectures/miw/KNN.pdf*, Metody Inteligencji Obliczeniowej - Metoda K Najbliższych Sąsiadów (KNN), Adrian Horzyk.

[4] *http://enroute.pl/knn-klasyfikacja/*, Informacje o sposobie działania algorytmu KNN.

[5] *http://thatdatatho.com/2018/10/04/cross-validation-the-wrong-way-right-way-feature-selection/*, Walidacja krzyżowa z wyborem cech - poprawnie i nie poprawnie.

[6] *https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html*, Wizualizacja zachowania walidacji krzyżowej w bibliotece `scikit-learn`.

[7] *https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html*, Informacje o dostępnych metrykach w dokumentacji biblioteki `scikit-learn`.

[8] *https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html*, Dokumentacja sposobu generowania macierzy pomyłek z wykorzystaniem biblioteki scikit-learn.

[9] *https://scikit-learn.org/stable/index.html*, Dokumentacja biblioteki `scikit-learn`.

[10] *Dealing with the evaluation of supervised classification algorithms*, Guzmán Santafé, Iñaki Inza, Jose Lozano, 2015.