

Data oddania: _____

Ocena: _____

Radosław Grela 216769

Jakub Wachała 216914

Zadanie 1: ekstrakcja cech, miary podobieństwa, klasyfikacja*

1. Cel

Celem naszego zadania było stworzenie aplikacji do klasyfikacji tekstów za pomocą metody k -NN (k najbliższych sąsiadów) oraz różnych metryk i miar podobieństwa, a następnie porównać kategorie z tymi wygenerowanymi przez aplikację.

2. Wprowadzenie

Głównym zagadnieniem projektowym, z którym mieliśmy do czynienia w ramach zadania 1 była klasyfikacja statystyczna tekstów na podstawie wektora wyekstrahowanych cech. Do przeprowadzenia eksperymentu zaimplementowaliśmy algorytm *k-najbliższych sąsiadów*.

Algorytm k -najbliższych sąsiadów (k -NN - *k-nearest neighbors*) to jeden z algorytmów zaliczanych do grupy algorytmów leniwych. Jest to taka grupa algorytmów, która szuka rozwiązania dopiero, gdy pojawia się wzorzec testujący. Przechowuje wzorce uczące, a dopiero później wyznacza się odległość wzorca testowego względem wzorców treningowych. [8]

Algorytm ten działa w taki sposób, że dla każdego wzorca testowego obliczana jest odległość za pomocą wybranej metryki względem wzorców treningowych, a następnie wybierana jest k najbliższych wzorców treningowych.

* Github: <https://github.com/Bonniu/KSR>

Wynik wyznaczony jest jako najczęstszy element wśród nich. W naszym zadaniu odległość ta jest równa skali podobieństwa tekstów.

2.1. Ekstrakcja cech

Do ekstrakcji cech charakterystycznych tekstu utworzyliśmy wektor cech, który opisuje tekst za pomocą 11 cech. Liczba słów zawsze jest liczona po zastosowaniu stop-listy oraz stemizacji, bez znaków przestankowych.

- C_1 - Stosunek słów kluczowych do wszystkich słów w pierwszych 10% tekstu. Obliczona jest za pomocą wzoru:

$$C_1 = s_{k10}/s_{10} \quad (1)$$

gdzie

s_{k10} - liczba słów kluczowych,

s_{10} - liczba wszystkich słów w pierwszych 10% tekstu.

Przed normalizacją cecha C_1 zawierała się w wartościach $\in [0, 1]$.

- C_2 - Stosunek słów kluczowych do wszystkich słów w ostatnich 10% tekstu. Obliczona jest za pomocą wzoru:

$$C_2 = s_{k90}/s_{90} \quad (2)$$

gdzie

s_{k90} - liczba słów kluczowych,

s_{90} - liczba wszystkich słów w ostatnich 10% tekstu.

Przed normalizacją cecha C_2 zawierała się w wartościach $\in [0, 0.5]$.

- C_3 - Stosunek słów kluczowych do wszystkich słów w dokumencie. Obliczona jest za pomocą wzoru:

$$C_3 = s_k/s \quad (3)$$

gdzie

s_k - liczba słów kluczowych,

s - liczba wszystkich słów w dokumencie.

Przed normalizacją cecha C_3 zawierała się w wartościach $\in [0, 0.155]$.

- C_4 - Stosunek słów kluczowych, których ilość liter $\in (0, 4]$ do wszystkich słów w dokumencie. Obliczona jest za pomocą wzoru:

$$C_4 = s_k/s \quad (4)$$

gdzie

s_k - liczba słów kluczowych, których ilość liter $\in (0, 4]$,

s - liczba wszystkich słów w dokumencie.

Przed normalizacją cecha C_4 zawierała się w wartościach $\in [0, 0.075]$.

- C_5 - Stosunek słów kluczowych, których ilość liter jest ≥ 8 do wszystkich słów w dokumencie. Obliczona jest za pomocą wzoru:

$$C_5 = s_k/s \quad (5)$$

gdzie

s_k - liczba słów kluczowych,

s - liczba wszystkich słów w dokumencie.

Przed normalizacją cecha C_5 zawierała się w wartościach $\in [0, 0.1]$.

- C_6 - Stosunek linii do ilości akapitów. Obliczona jest za pomocą wzoru:

$$C_6 = l/a \quad (6)$$

gdzie

l - liczba linii,

a - liczba akapitów.

Przed normalizacją cecha C_6 zawierała się w wartościach $\in [1, 14]$.

- C_7 - Stosunek słów, których ilość liter jest większa niż 6 do wszystkich słów. Obliczona jest za pomocą wzoru:

$$C_7 = s_6/s \quad (7)$$

gdzie

s_6 - liczba słów których ilość liter jest większa niż 6,

s - liczba wszystkich słów w dokumencie.

Przed normalizacją cecha C_7 zawierała się w wartościach $\in [0, 0.591]$.

- C_8 - Stosunek słów kluczowych, których ilość liter jest ≤ 6 do wszystkich słów w dokumencie. Obliczona jest za pomocą wzoru:

$$C_8 = s_{6m}/s \quad (8)$$

gdzie

s_{6m} - liczba słów kluczowych, których ilość liter jest ≤ 6 ,

s - liczba wszystkich słów w dokumencie.

Przed normalizacją cecha C_8 zawierała się w wartościach $\in [0.409, 1]$.

- C_9 - Ilość słów unikalnych. Jest to liczba słów, które wystąpiły w tekście co najmniej raz. Przykładowo, dla zdania „*Być albo nie być*” ilość słów unikalnych jest równa 3 (*być, albo, nie*).

Przed normalizacją cecha C_9 przyjmuje wartości $\in [1, 420]$.

- C_{10} - Ilość słów, których ilość liter $\in [5, 8]$. Pseudokod obliczający wartość cechy C_{10} :

— $C_{10}=0$

— Dla każdego słowa w artykule:

— Jeżeli $\text{długość słowa} \geq 5$ i $\text{długość słowa} \leq 8$:

— $C_{10}++$;

— Zwróć C_{10}

Przed normalizacją cecha C_{10} zawierała się w wartościach $\in [1, 574]$.

- C_{11} - Najczęściej występujące słowo kluczowe. Jest to cecha tekstowa, której podobieństwo z innym słowem mierzymy jedną z dwóch miar opisanych w sekcji *Metryki i miary podobieństwa*.

2.2. Wyznaczanie słów kluczowych

Wyznaczenie słów kluczowych przebiega w następujący sposób: na początek za pomocą klasy WordCounter zliczane są wszystkie słowa w artykułach oraz jednocześnie dodawane do odpowiednich list w tej klasie. Każda zmienna jest listą stringów o nazwie wordCountDictionary + nazwa kraju. Dodatkowo, przechowywany jest słownik typu $\langle \text{string}, \text{int} \rangle$, którego kluczem jest słowo, a wartością to ilość wystąpień tego słowa we wszystkich artykułach.

Po podliczeniu wszystkich słów oraz przydzieleniu do odpowiednich list wybieramy po 18 najpopularniejszych słów dla każdego kraju, które występują tylko w tym jednym konkretnym kraju. Na koniec $18 * 6 = 108$ słów zostaje słowami kluczowymi. Cały proces wyznaczania słów kluczowych jest dokonywany po zastosowaniu stop-listy oraz po stemizacji. Ponadto, proces wybierania słów kluczowych pomija 20% wszystkich podliczonych słów, aby proces dopasowywania słów kluczowych do krajów nie trwał zbyt długo.

2.3. Metryki i miary podobieństwa

Do liczenia odległości pomiędzy artykułami oraz obliczenia miary podobieństwa używaliśmy 3 metryk i 2 miar podobieństwa ciągów tekstowych.

1. Metryka Euklidesowa - aby obliczyć odległość $d_e(x, y)$ między wektorami x i y należy obliczyć pierwiastek kwadratowy z sumy kwadratów różnic wartości współrzędnych wektora o tych samych indeksach. Wzór jest następujący [5]:

$$d_e(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \quad (9)$$

gdzie x_i i y_i to cechy wektora.

2. Metryka Manhattan - odległość $d_m(x, y)$ jest równa sumie wartości bezwzględnych z różnic wartości współrzędnych wektora o tych samych indeksach [3]:

$$d_m(x, y) = \sum_{n=1}^N |x_n - y_n| \quad (10)$$

gdzie x_i i y_i to cechy wektora.

3. Metryka Czebyszewa - odległość $d_c(x, y)$ w tej metryce jest równa maksymalnej wartości bezwzględnych różnic współrzędnych punktów x oraz y , zgodnie ze wzorem [4]:

$$d_c(x, y) = \max_i |x_i - y_i| \quad (11)$$

gdzie x_i i y_i to cechy wektora.

4. Miara n -gramów - metoda ta określa podobieństwo łańcuchów tekstowych s_1, s_2 w oparciu o ilość wspólnych podciągów n -elementowych, czyli n -gramów [2]:

$$sim_n(s_1, s_2) = \frac{1}{N - n + 1} \sum_{i=1}^{N-n+1} h(i) \quad (12)$$

gdzie

$h(i) = 1$, jeśli n -elementowy podciąg zaczynający się od i -tej pozycji w s_1 występuje co najmniej raz w s_2 , w przeciwnym razie $h(i) = 0$

$N - n + 1$ - ilość możliwych n -elementowych podciągów w s_1 .

W naszym programie n jest stałe i wynosi 3.

5. Uogólniona miara n -gramów (Miara Niewiadomskiego) - ta miara jest ulepszoną wersją miary n -gramów. Bada ona podobieństwo poprzez spraw-

dzenie podciągów różnej długości od jedno- do N -elementowych, gdzie N jest długością słowa [2]:

$$\mu_N(s_1, s_2) = \frac{2}{N^2 + N} \sum_{i=1}^{N(s_1)} \sum_{j=1}^{N(s_1)-i+1} h(i, j) \quad (13)$$

gdzie

$h(i, j) = 1$, jeśli i -elementowy podciąg w słowie s_1 zaczynający się od j -tej pozycji w słowie s_1 pojawia się co najmniej raz w słowie s_2 , w przeciwnym razie $h(i, j) = 0$

$N(s_1), N(s_2)$ – ilość liter w słowach s_1 i s_2 ;

$N = \max\{N(s_1), N(s_2)\}$

$\frac{N^2+N}{2}$ - ilość możliwych podciągów od 1-elementowych do N -elementowych w słowie o długości N .

2.4. Miary jakości

W wynikach klasyfikacji używamy następujących miar jakości [9]:

- Accuracy:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

- Precision

$$PPV = \frac{TP}{TP + FP} \quad (15)$$

- Recall

$$TPR = \frac{TP}{TP + FN} \quad (16)$$

Oznaczenia użytych symboli:

TP - miara prawdziwie pozytywna (*true positive*)

TN - miara prawdziwie negatywna (*true negative*)

FP - miara fałszywie pozytywna (*false positive*)

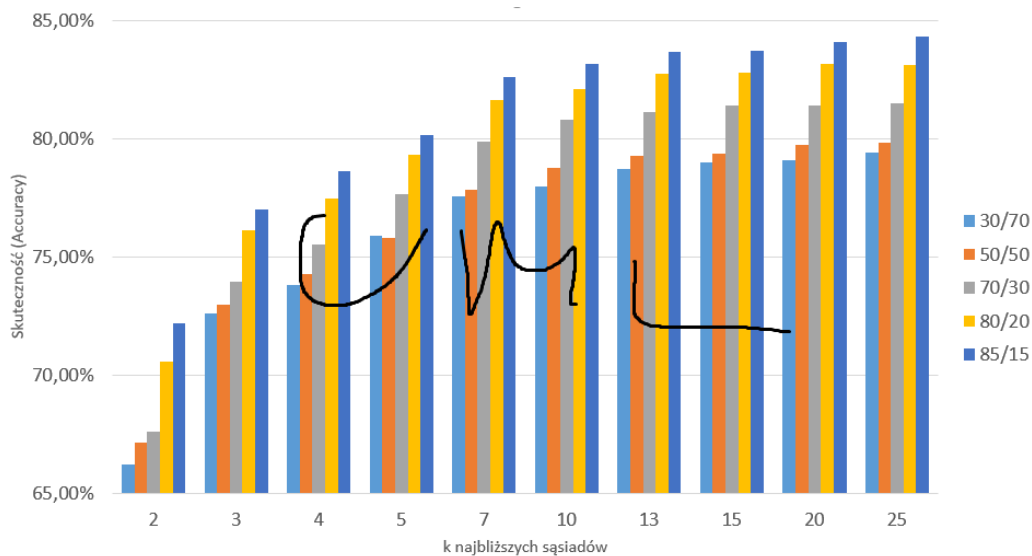
FN - miara fałszywie negatywna (*false negative*)

3. Opis implementacji

Nasza aplikacja została utworzona w języku C# i jest to aplikacja konsolowa. Poniżej opisane zostały wszystkie klasy oraz dane zawarte w naszym projekcie:

- Klasa Program to klasa główna naszego programu. Jest swego rodzaju kontrolerem dla pozostałych klas. Znajduje się tutaj funkcja *main*, która rozpoczyna wykonywanie programu.
- W katalogu *dane* znajdują się wszystkie pliki z artykułami, które są wykorzystywane do badań.
- Klasa Metric jest klasą abstrakcyjną. Odpowiada za obliczenia odległości tekstów. Po tej klasie dziedziczą klasy: EuclideanMetric, ChebyshevMetric oraz ManhattanMetric.

- Klasa Measure jest klasą abstrakcyjną. Po niej dziedziczą klasy *GeneralizedNGramsMeasure* i *NGramsMeasure*, które odpowiadają za obliczanie miar podobieństwa łańcuchów tekstowych.
 - Klasa Feature jest klasą abstrakcyjną. Po niej dziedziczy 10 klas: Feature 1-10, które reprezentują każdą z 10 wyekstrahowanych przez nas cech.
 - Klasa Stemmer to klasa, która odpowiada za stemizację tekstów. Została ona zapożyczona z [6]
 - Klasa StopwordTool jest klasą odpowiedzialną za usuwanie słów znajdujących się na stopliście. Również została znaleziona i zapożyczona z Internetu ze strony [7]
 - WordCounter jest używany do zliczania słów wszystkich artykułów i podania ich liczności. Potrzebny głównie do wyznaczenia słów kluczowych.
 - Klasa KeyWords odpowiada za wyznaczenie 100 słów kluczowych. Metoda wyznaczania słów kluczowych została opisana w sekcji 2.
 - Klasa FileReader odpowiada za otwieranie każdego pliku z artykułami
 - FileParser to klasa odpowiedzialna za parsowanie danych z konkretnego pliku.
 - Article to klasa reprezentująca artykuł. Zawiera takie cechy jak: tekst oryginalny, tekst przetworzony, *place*, *classifiedPlace*, wektor cech.
 - Klasa Neighbor to klasa, która przechowuje artykuł oraz obliczoną wartość algorytmu k-NN dla konkretnego, obecnie sprawdzanego artykułu w algorytmie. Wykorzystujemy ją, aby znaleźć najbliższych k sąsiadów.
 - KNN to klasa odpowiedzialna za algorytm k najbliższych sąsiadów.
- Na rysunku 1 przedstawiony został diagram UML naszego programu.



Rysunek 1. Diagram UML.

4. Materiały i metody

Wykonana przez nas klasyfikacja została wykonana za pomocą wszystkich trzech metryk oraz dwóch miar podobieństwa. Każdy przypadek testowy był

klasyfikowany dla dziesięciu różnych wartości k najbliższych sąsiadów: 2, 3, 4, 5, 7, 10, 13, 15, 20, 25.

Klasyfikacji dokonywaliśmy tylko na tych tekstach, które miały jedną z etykiet: *west-germany*, *usa*, *france*, *uk*, *canada*, *japan* i były to ich jedyne etykiety.

Dokonaliśmy pięciu różnych podziałów na dane testowe oraz treningowe:

- 30% dane treningowe, 70% dane testowe
- 50% dane treningowe, 50% dane testowe
- 70% dane treningowe, 30% dane testowe
- 80% dane treningowe, 20% dane testowe
- 85% dane treningowe, 15% dane testowe

5. Wyniki

5.1. Badanie podziału na dane treningowe i testowe

k	Accuracy [%]
2	66,19
3	72,61
4	73,81
5	75,87
7	77,56
10	77,95
13	78,71
15	78,98
20	79,07
25	79,41

Tabela 1. Skuteczność klasyfikacji dla metryki Euklidesowej dla podziału zbioru 30% treningowe/70% testowe.

k	Accuracy [%]
2	67,14
3	72,97
4	74,25
5	75,79
7	77,82
10	78,74
13	79,26
15	79,35
20	79,74
25	79,81

Tabela 2. Skuteczność klasyfikacji dla metryki Euklidesowej dla podziału zbioru 50% treningowe/50% testowe.

k	Accuracy [%]
2	67,62
3	73,95
4	75,52
5	77,64
7	79,86
10	80,77
13	81,11
15	81,38
20	81,38
25	81,48

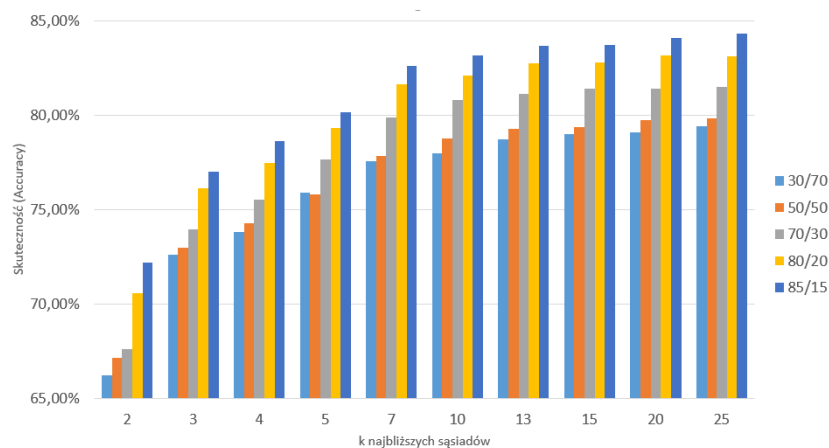
Tabela 3. Skuteczność klasyfikacji dla metryki Euklidesowej dla podziału zbioru 70% treningowe/30% testowe.

k	Accuracy [%]
2	70,56
3	76,11
4	77,47
5	79,31
7	81,62
10	82,10
13	82,73
15	82,80
20	83,17
25	83,09

Tabela 4. Skuteczność klasyfikacji dla metryki Euklidesowej dla podziału zbioru 80% treningowe/20% testowe.

k	Accuracy [%]
2	72,16
3	76,99
4	78,60
5	80,16
7	82,59
10	83,13
13	83,67
15	83,72
20	84,06
25	84,30

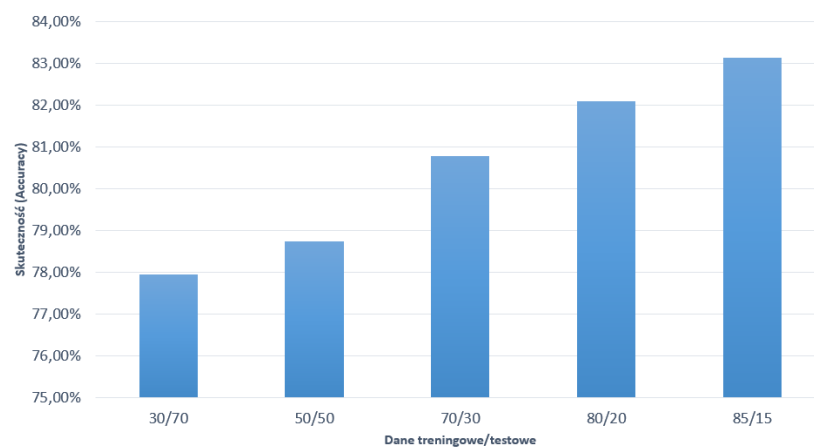
Tabela 5. Skuteczność klasyfikacji dla metryki Euklidesowej dla podziału zbioru 85% treningowe/15% testowe.



Rysunek 2. Dane z tabel 1-5 zebrane na wykresie.

Dane treningowe/testowe	Accuracy [%]
30/70	77,95
50/50	78,74
70/30	80,77
80/20	82,10
85/15	83,13

Tabela 6. Zależność Accuracy od pięciu wartości proporcji podziału zbioru dla $k=10$.

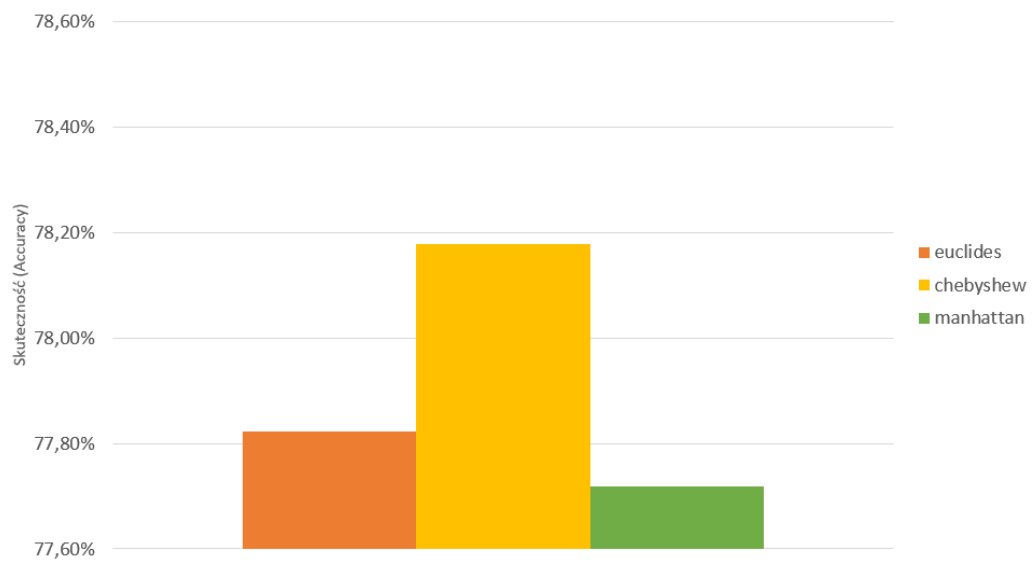


Rysunek 3. Wykres przedstawiający zależność Accuracy od pięciu wartości proporcji podziału zbioru dla $k=10$.

5.2. Badanie różnic w metrykach

Metryka	Accuracy [%]
Euklidesowa	77,82
Czebyszewa	78,18
Manhattan	77,72

Tabela 7. Zależność Accuracy od wyboru metryki dla $k=7$ i podziału 50/50.

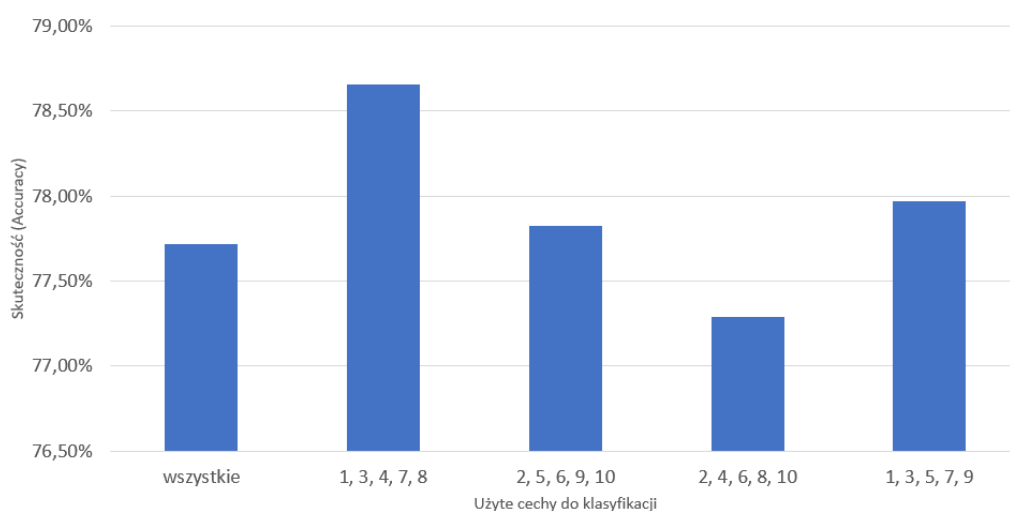


Rysunek 4. Wykres przedstawiający zależność Accuracy od wyboru metryki dla $k=7$ i podziału 50/50.

5.3. Badanie różnic w wyborze podzbioru cech

Podzbiór cech	Accuracy [%]
Wszystkie cechy	77,72
C_1, C_3, C_4, C_7, C_8	78,65
$C_2, C_5, C_6, C_9, C_{10}$	77,82
$C_2, C_4, C_6, C_8, C_{10}$	77,29
C_1, C_3, C_5, C_7, C_9	77,97

Tabela 8. Zależność Accuracy od wyboru podzbioru cech dla $k=7$ i podziału 50/50.



Rysunek 5. Wykres przedstawiający zależność Accuracy od wyboru podzbioru cech dla $k=7$ i podziału 50/50.

6. Dyskusja

Sekcja ta powinna zawierać dokładną interpretację uzyskanych wyników eksperymentów wraz ze szczegółowymi wnioskami z nich płynącymi. Najcenniejsze są, rzecz jasna, wnioski o charakterze uniwersalnym, które mogą być istotne przy innych, podobnych zadaniach. Należy również omówić i wyjaśnić wszystkie napotkane problemy (jeśli takie były). Każdy wniosek powinien mieć poparcie we wcześniej przeprowadzonych eksperymentach (odwołania do konkretnych wyników). Jest to jedna z najważniejszych sekcji tego sprawozdania, gdyż prezentuje poziom zrozumienia badanego problemu.

7. Wnioski

W tej, przedostatniej, sekcji należy zamieścić podsumowanie najważniejszych wniosków z sekcji poprzedniej. Najlepiej jest je po prostu wypunktować.

Znów, tak jak poprzednio, najistotniejsze są wnioski o charakterze uniwersalnym.

Literatura

- [1] Niewiadomski, Adam. Methods for the Linguistic Summarization of Data: Applications of Fuzzy Sets and Their Extensions. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2008. ISBN 978-83-60434-40-6
- [2] https://ftims.edu.p.lodz.pl/pluginfile.php/132368/mod_folder/content/0/ksr-wyklad-2009.pdf?forcedownload=1 [dostęp 22.03.2020]
- [3] https://en.wikipedia.org/wiki/Taxicab_geometry [dostęp 01.04.2020]
- [4] https://en.wikipedia.org/wiki/Chebyshev_distance [dostęp 01.04.2020]
- [5] https://en.wikipedia.org/wiki/Euclidean_distance [dostęp 01.04.2020]
- [6] <https://tartarus.org/martin/PorterStemmer/csharp.txt> [dostęp 22.03.2020]
- [7] <https://www.dotnetperls.com/stopword-dictionary> [dostęp 22.03.2020]
- [8] <http://home.agh.edu.pl/~horzyk/lectures/miw/KNN.pdf> [dostęp 22.03.2020]
- [9] https://pl.wikipedia.org/wiki/Tablica_pomy%C5%82ek [dostęp 01.04.2020]