Dokumentacja projektu z MED - algorytm NBC

Łukasz Reszka (300257), Kamil Wojciechowski (zrezygnował)

17.01.2023 r.

Spis treści

1. Etap I

Etap I projektu koncentruje się w dużej mierze wokół implementacji algorytmu NBC i przetestowania jego działania. Najpierw jednak należało sformułować problem, który algorytm NBC umiałby rozwiązać. Wymagało to wyszukania źródeł opisujących NBC i odpowiednie zbioru danych eksperymentalnych. Następnie zdefiniowano plan realizacji całego projektu i eksperymenty do przeprowadzenia. Zidentyfikowano również wszystkie potrzebne na poziomie implementacji narzędzia. Dodatkowo, wybrano algorytm kandydujący do bycia realizowanym w etapie II. Końcowa część etapu I to głównie prezentacja wyników badań i wyciągniętych wniosków.

1.1. Opis tematu projektu

Celem realizowanego tematu jest opracowanie modelu grupującego artykuły ze strony BBC (rok 2004-2005) względem dziedziny, której one dotyczą. W celu przeprowadzenia grupowania wiadomości zostanie wykorzystany algorytm NBC [1] (z ang. *Neighborhood-Based Clustering Algorithm*), którego implementacja i przetestowanie stanowi zasadniczą część projektu. W drugiej fazie realizacji zadania, algorytm NBC zostanie przekształcony do formy algorytmu TI-NBC [2], którego cechuje wyższa efektywność grupowania danych i inny sposób wyznaczania indeksów. Skuteczność obu algorytmów grupowania zostanie ze sobą porównana.

1.2. Algorytm NBC

Klastrowanie w eksploracji danych. Klastrowanie jest jedną ze znanych i często stosowanych metod eksploracji danych. Jego celem jest przypisanie obiektów danych (lub punktów) do różnych klastrów w taki sposób, aby obiekty, które zostały przypisane do tych samych klastrów, były bardziej podobne do siebie niż do obiektów przypisanych do innych klastrów. Można powiedzieć, że proces klastrowania w eksploracji danych polega zasadniczo na składaniu zestawu abstrakcyjnych obiektów w grupy podobnych obiektów. Algorytmy klastrowania mogą działać na różnych typach danych, takich jak bazy danych, wykresy, multimedia oraz tekst.

W przypadku **klastrowania tekstu**, główną ideą jest to, że dokumenty mogą być reprezentowane numerycznie jako wektory cech. Podobieństwo w tekście można porównać, mierząc odległość między tymi wektorami cech. Obiekty znajdujące się blisko siebie powinny należeć do tego samego klastra. Obiekty oddalone od siebie powinny należeć do różnych klastrów. Przykłady użycia grupowania tekstu:

- Generowanie taksonomii: Automatyczne generowanie hierarchicznych taksonomii do przeglądania treści.
- Identyfikacja fałszywych wiadomości: wykrywanie, czy wiadomość jest prawdziwa, czy fałszywa.
- Tłumaczenie językowe: Tłumaczenie zdania z jednego języka na inny. Filtrowanie spamu: wykrywa niechciane e-maile/wiadomości.
- Analiza problemów z obsługą klienta: Identyfikacja problemów często zgłaszanych do pomocy technicznej.

Algorytm NBC [1] (z ang. *Neighborhood-Based Clustering Algorithm*) jest algorytmem grupującym gęstościowo dane. Grupy tworzone są w oparciu o gęstość podprzestrzeni, wyrażonej **współczynnikiem gęstości NDF**. Na podstawie wartości tego współczynnika wyróżniane są punkty rdzeniowe, które wraz ze swoim **k^+-sąsiedztwem** stanowią gęstą przestrzeń - przestrzeń, którą można uznać za grupę lub część grupy.

Współczynnik gęstości NDF (z ang. *Neighborhood Density Factor*) jest wyliczany jako stosunek liczności odwrotnego k^+-sąsiedztwa do liczności k^+-sąsiedztwa. Zgodnie z definicją, **k^+-sąsiedztwo** punktu *p* jest zbiorem wszystkich punktów różnych od *p*, których odległość do punktu *p* nie przekracza odległości jego dowolnego najdalszego k-tego sąsiada do punktu *p*. Z kolei **odwrotne k^+-sąsiedztwo** punktu *p* jest zbiorem punktów, dla których *p* jest k^+-sąsiadem. Współczynnik gęstości NDF wyraża więc relatywną, lokalną gęstość danej podprzestrzeni. Tak wyrażona gęstość umożliwia algorytmowi NBC wyznaczanie grup o różnorakiej gęstości / granularności, co stanowi wyzwanie dla innych algorytmów np.: DBSCAN.

W ramach projektu zostanie porównana skuteczność podstawowego algorytmu NBC z jego zmodyfikowaną wersję TI-NBC (opisaną w następnym rozdziale). Implementacja obu algorytmów powstanie w oparciu o [11]. Poniżej zamieszczono **pseudokod algorytmu NBC**:

ALGORYTM NBC - PSEUDOKOD

```
algorytm_NBC (zbiór_punktów, k) {
    dla każego punkt z zbiór_punktów {
        punkt.grupa = nieokreślona
    }
    oblicz_współczynnik_NDF(zbiór_punktów, k)
    numer_tworzonej_grupy = 0
    dla każdego punkt z zbiór_punktów {
        //jeśli punkt jest punktem rdzeniowym
        jeśli punkt.grupa == nieokreślona oraz punkt.NFD >= 1 {
            punkt.grupa = numer_tworzonej_grupy
            ziarno = pusty_zbiór()
            dla każdego punkt2 z k+-sąsiedztwo(punkt) {
                punkt2.grupa = numer_tworzonej_grupy
                jeśli punkt2.NDF >= 1 {
                    ziarno = ziarno + punkt2
```

```
            }
        }
        dopóki ziarno =/= pusty_zbiór() {
            punkt_ziarna = zwróć_punkt(ziarno)
            dla każdego punkt3 z k+-sąsiedztwo(punkt_ziarna) {
                jeśli punkt3.grupa == nieokreślona {
                    punkt3.grupa = numer_tworzonej_grupy
                    jeśli punkt3.NDF >= 1 {
                        ziarno = ziarno + punkt3
                    }
                }
            }
            ziarno = ziarno - punkt_ziarna
        }
        numer_tworzonej_grupy = numer_tworzonej_grupy + 1
    }
}
dla każdego punkt4 z zbiór_punktów {
    jeśli punkt4.grupa == nieokreślona {
        punkt4.grupa = punkt_szumu
    }
}
```

1.3. Algorytm TI-NBC

Dla efektywności działania algorytmu NBC, kluczowe jest sprawne wyznaczanie k^+-sąsiedztw poszczególnym punktów. W tym celu korzysta się z indeksów przestrzennych - typu **plik siatkowy** czy **R*-drzewo**. W implementacji wykorzystano strukturę ***ball tree*** (dzielącą przestrzeń na hipersfery). Istnieje też nowatorskie podejście do kwestii indeksu, opisane w publikacji [2]. Przedstawiono w niej **algorytm TI-NBC**, który wylicza indeks w oparciu o **nierówność trójkąta**. Ten sposób wyznaczania k^+-sąsiedztw okazuje się bardziej efektywny niż zastosowanie tradycyjnych indeksów przestrzennych.

1.4. Wykorzystany w zadaniu zbiór danych

W projekcie wykorzystano zbiór danych dostępny pod adresem [5]. Składa się on z 2225 artykułów BBC, które były zamieszczane na stronie internetowej BBC w latach 2004-2005. Co ważne, każdy artykuł jest przypisany do jednej z pięciu kategorii: *biznes, rozrywka, polityka, sport* i *technologia*. Dzięki temu możliwe będzie skorzystanie z zewnętrznych miar jakości przeprowadzonego grupowania. Pierwotnie każdy artykuł znajduje się w osobnym pliku tekstowym; pliki te są umieszczone w odpowiednim folderze, odpowiadającym kategorii artykułu.

1.5. Plan realizacji projektu

- Pobranie zbioru artykułów BBC z [5]
- Przetworzenie zebranych danych (*preprocessing*):
    a. zamiana na małe litery
    b. usunięcie znaczników html, liczb, znaków interpunkcyjnych i diaktrycznych
    c. usunięcie zbędnych białych znaków
    d. usunięcie słów ze *stop listy* (z **nltk** dla języka angielskiego)
    e. stemming - wyodrębnienie rdzeni wyrazów
    f. wektoryzacja - przeniesienie danych tekstowych do przestrzeni liczb
- Grupowanie artykułów pod względem tematyki - zastosowanie zaimplementowanych algorytmów
- Przeprowadzenie testów, sprawdzenie jakości grupowania
- Wizualizacja i analiza wyników (zastosowanie techniki ***PCA*** [7] do redukcji wymiarów przy generowaniu wykresów 2D)

1.6. Wybrane narzędzia i biblioteki

- język **Python 3.9**
- **PyCharm IDE**
- **Git** i wydziałowy **Gitlab**
- bibliotekę **pandas** do manipulowania danymi
- bibliotekę **nltk**, **scikit-learn**, **Unidecode, numpy** i **torchtext** m.in. do przetwarzania języka naturalnego
- bibliotekę **matplotlib** i **seaborn** do wizualizacji
- bibliotekę **tqdm** do wyświetlania pasków postępu

1.7. Zaplanowane eksperymenty

W ramach eksperymentów zostanie porównana jakość grupowania przeprowadzonego przez algorytm NBC i TI-NBC, względem referencyjnego podziału na grupy. W tym celu zostaną wykorzystane różne miary ewaluacji grupowania - zewnętrzne (np.: *Rand*) oraz wewnętrzne (np.: *wskaźnik sylwetkowy*). Przypisanie artykułów do klastrów zostanie zwizualizowane na odpowiednich wykresach.

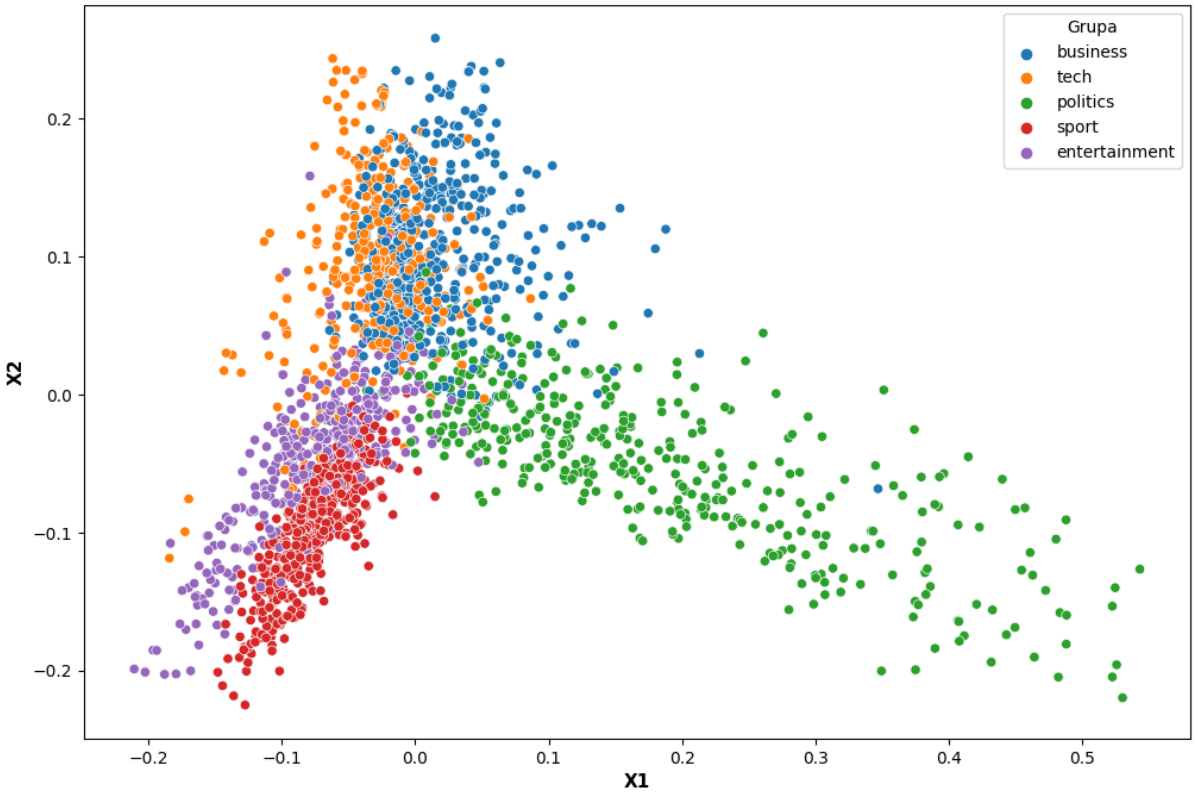
Oprócz jakości grupowania, zostanie zbadana i porównana efektywność działania obu zaimplementowanych algorytmów. Zostanie zweryfikowana też poprawność samej implementacji na przykładowym zbiorze testowym - punktów w przestrzeni 2D.

1.8. Rezultaty przeprowadzonych testów

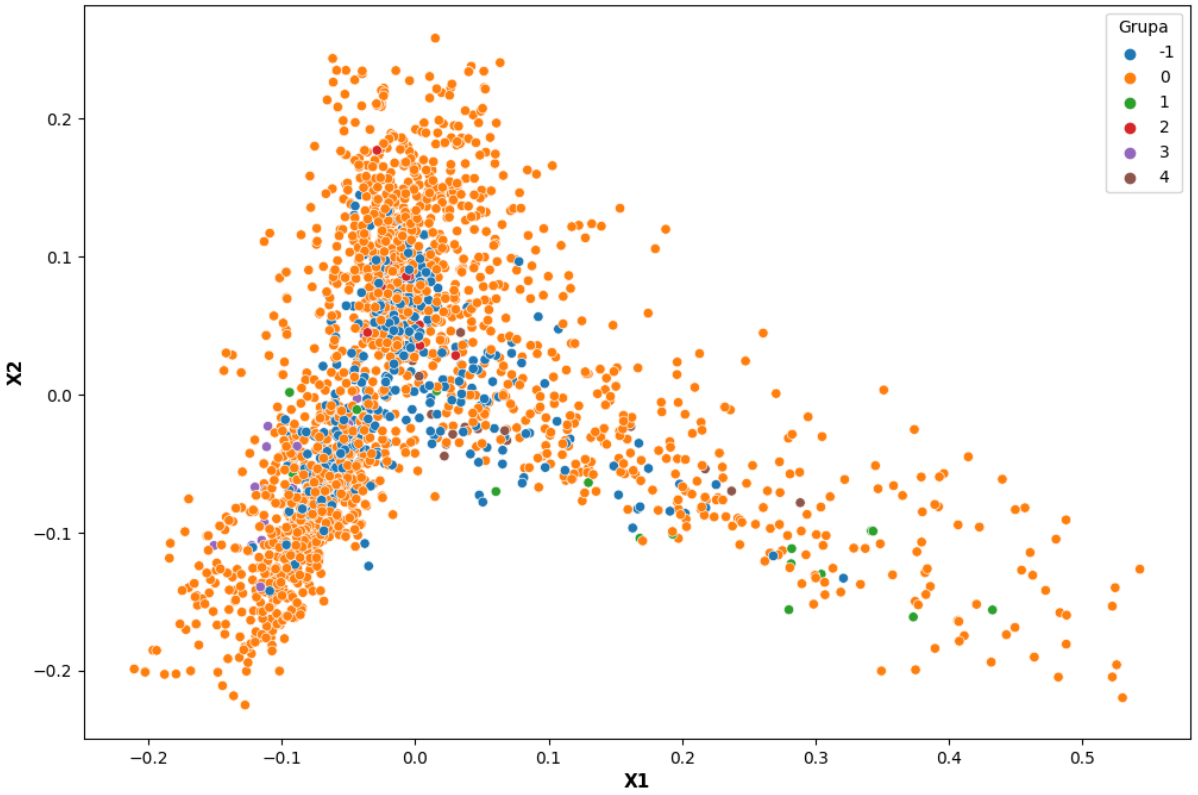
W pierwotnej implementacji algorytmu NBC, do zakodowania artykułów w postaci wektorów liczb zastosowano metodę ***TF-IDF*** (z ang. *Term Frequency - Inverse Document Frequency*). Współrzędne wektorów odpowiadały częstości występowania poszczególnych wyrazów w artykułach. Przy wyliczaniu tych częstości wyważano znaczenie lokalne termu i jego znaczenie w kontekście globalnym (wszystkich artykułów). Poniżej zestawiono ze sobą referencyjny podział artykułów na grupy z efektem działania zaimplementowanego algorytmu **NBC**. Rysunek drugi uzyskano przyjmując parametr ***k* = 17** i ograniczając ilość branych pod uwagę **wyrazów** do jedynie **10 000** najbardziej znaczących (tym samym ograniczając wymiar przestrzeni wektorów). Wartości parametrów dobrano eksperymentalnie tak, aby uzyskać możliwie najlepszy rezultat grupowania.

Nietrudno zauważyć, że jakość grupowania przeprowadzonego przez algorytm NBC pozostawia wiele do życzenia. Co prawda, algorytm wyróżnił odpowiednią liczbę klastrów (nie licząc grupy ***-1*** - **punktów szumu**), ale niemal wszystkie artykuły zaklasyfikował do tej samej grupy. Dążąc do poprawy jakości grupowania, zdecydowano się na wybór innej metody wektoryzacji danych tekstowych, która bardziej brałaby pod uwagę kontekst i semantykę wyrazów przy przypisywaniu wektorów.

Klastry referencyjne (TF-IDF)

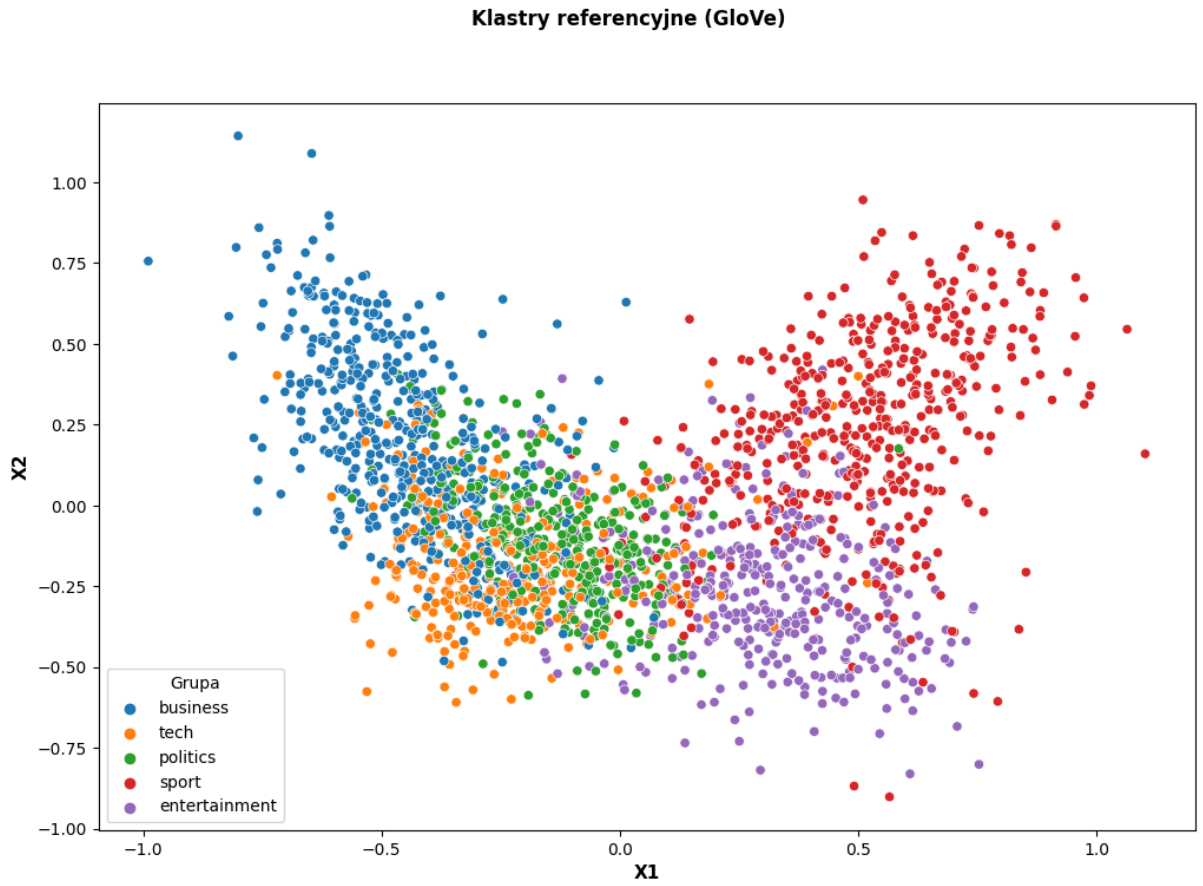


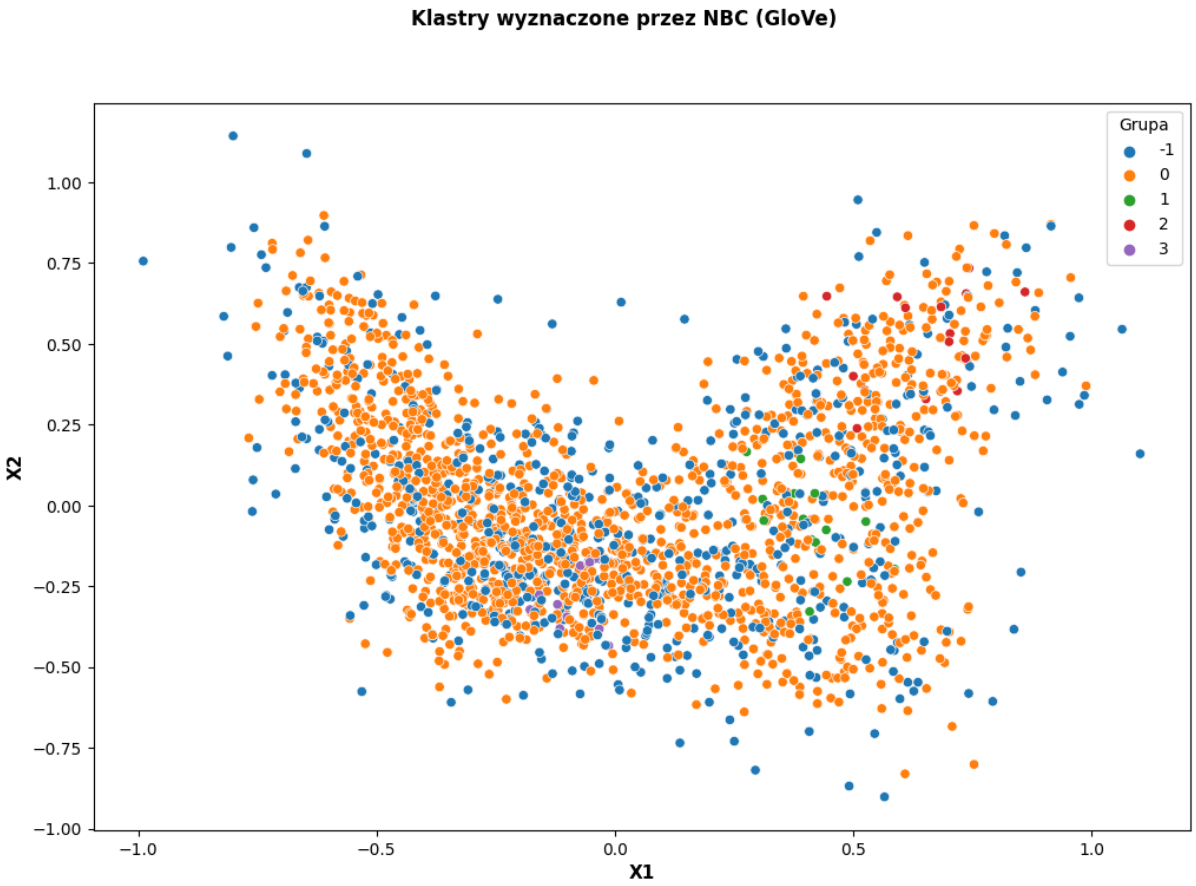
Klastry wyznaczone przez NBC (TF-IDF)



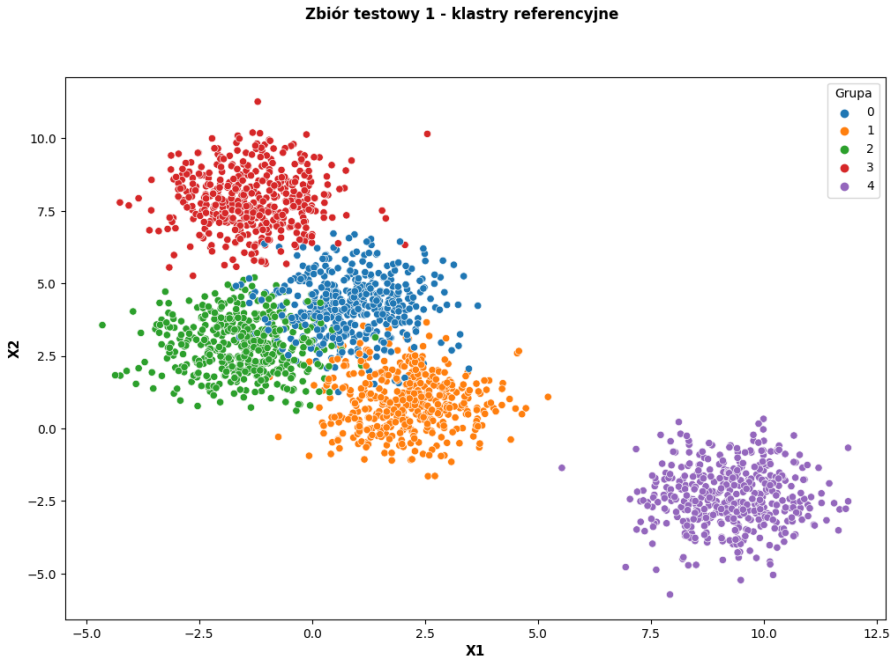
W drugim podejściu wykorzystano model osadzania słów ***GloVe*** (z ang. *Global Vectors for Word Representation*) przy wektoryzacji danych tekstowych. W tym celu pobrano już gotowy zbiór przypisań słów do wektorów ze strony [4]. Zbiór ten został wyznaczony właśnie przy użyciu modelu *GloVe* na 6 milionach słów pochodzących z Wikipedii z 2014 roku. *GloVe* każde słowo przedstawił jako odpowiadający mu w przestrzeni wektor liczb, w taki sposób, że słowa podobne znaczeniowo znajdują się blisko siebie w tej przestrzeni. Przy reprezentacji poszczególnych artykułów odnajdywano więc reprezentację liczbową poszczególnych słów artykułu, a potem wyciągano średnią ze wszystkich wektorów liczb.

Poniżej zamieszczono efekt działania algorytmu NBC przy użyciu modelu osadzania słów *GloVe* w porównaniu z klastrami referencyjnymi. Rysunek przedstawia efekt wywołania algorytmu NBC dla parametrów ***k* = 14** i **wymiar wektora liczb = 100**. Ich wartości zostały dobrane eksperymentalnie tak, aby zmaksymalizować jakość grupowania. Niestety, pomimo zastosowania modelu *GloVe* jakość grupowania w dalszym ciągu nie była zadowalająca.

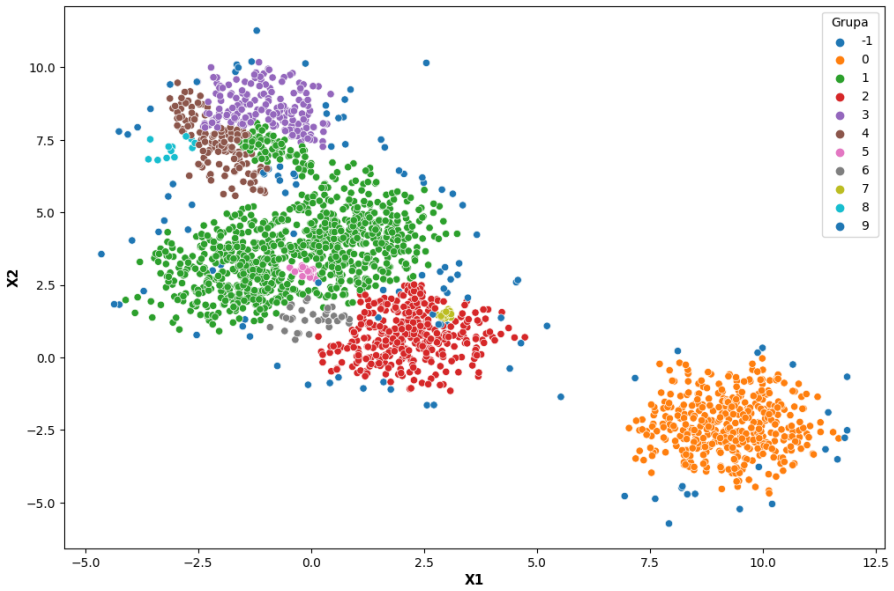




W celu sprawdzenia poprawności implementacji algorytmu NBC, przetestowano jego działanie na dwóch zbiorach testowych, wygenerowanych automatycznie ([3]). Poniższe wykresy potwierdzają, że zaimplementowany algorytm faktycznie jest w stanie wynajdywać odpowiednie grupy danych, z dokładnością zależną od samego rozłożenia punktów w przestrzeni liczbowej.

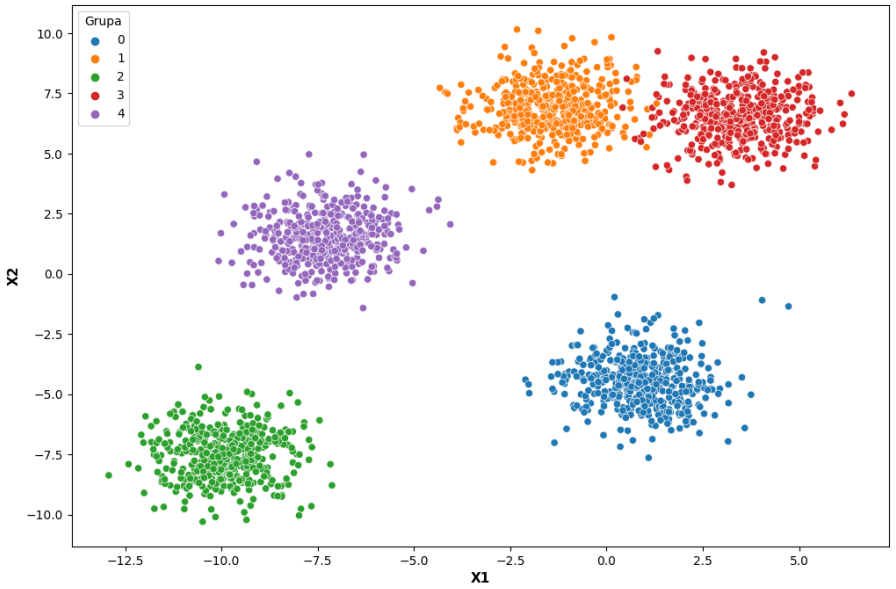


Zbiór testowy 1 - klastry NBC



Dla parametru k = 9.

Zbiór testowy 2 - klastry referencyjne



Dla k = 20

1.9. Podsumowanie I etapu

Algorytm NBC na wybranym zbiorze danych (artykułów BBC) nie dokonał wysokiej jakości grupowania, zgodnego z wzorcowym podziałem. W ramach projektu badano efekty działania algorytmu dla różnych wartości *k* (liczba sąsiadów), wartości wymiarów wektorów liczbowych, reprezentujących teksty, oraz technik wektoryzacji słów (*TF-IDF* i *GloVe*). Działanie te nie spowodowały znaczącej zmiany ewaluacji grupowania. Z tego powodu, nie było potrzeby wyznaczać miar zewnętrznych i wewnętrznych grupowania. Ponadto, działanie samego algorytmu NBC na danych testowych (automatycznie wygenerowanych) potwierdziło jego skuteczność.

Na podstawie powyższego, nie można ocenić skuteczności działania algorytmu NBC. Aby uzyskać pewność, co do funkcjonowania algorytmu NBC, można by rozważyć wybór innego zbioru danych eksperymentalnych lub wykorzystanie innego algorytmu grupującego gęstościowo oraz w inny sposób (poprzez porównanie jego wyników z algorytmem NBC).

2. Etap II

Pierwotnym celem etapu II było zaimplementowanie algorytmu TI-NBC i porównanie go z algorytmem NBC pod względem wydajności i jakości grupowania. Jednakże, biorąc pod uwagę nieplanowane zmiany w zespole projektowym oraz niską jakość grupowania przeprowadzonego przez algorytm NBC na wybranym zbiorze danych, zmieniono pierwotny zamysł. Ostatecznie zdecydowano się na porównanie działania algorytmu NBC z innymi algorytmami służącymi do grupowania. Badania przeprowadzono na tym samym zbiorze danych, który był wykorzystany w etapie I. W ramach zestawienia ze sobą algorytmów, wykonano wizualizację efektów ich działania oraz obliczono odpowiednie miary jakości grupowania.

2.1. Wybrane algorytmy grupowania

Efekty grupowania algorytmu NBC zostały zestawione z efektami działania następujących algorytmów:

- **Agglomerative Clustering** - **grupowanie hierarchiczne w podejściu aglomeracyjnym**. Algorytm polega na sekwencyjnym grupowaniu obiektów, podczas którego powstaje zagnieżdżona hierarchia klastrów zwana *dendrogramem*. Początkowo każdy obiekt przestrzeni poszukiwań stanowi osobny klaster. Następnie, w kolejnych iteracjach najbliższe klastry są ze sobą łączone. Działanie algorytmu kończy się przy osiągnięciu zadanej liczby klastrów.

- Algorytm **k-średnich** (z ang. *k-means*) - **grupowanie iteracyjno-optymalizacyjne**. Algorytm tworzy jeden podział zbioru obiektów zamiast hierarchicznej struktury. Na początku wybiera losowo *k* punktów jako środki klastrów. Obiekty są przypisywane do tego klastra, do którego środka mają najbliżej. Następnie wyliczana jest optymalizowana funkcja kryterialna. W kolejnych iteracjach położenia środków klastrów i przypisane do nich punktu są uaktualniane tak, by minimalizować funkcję kryterialną. Działanie algorytmu kończy działanie w momencie gdy nie następuje przemieszczanie się obiektów pomiędzy klastrami przez pewnę liczbę iteracji.

- Algorytm **OPTICS** (z ang. *Ordering Points To Identify the Clustering Structure*) - **grupowanie gęstościowe**. Algorytm stanowi rozszerzenie algorytmu DBSCAN, ale w przeciwieństwie do DBSCAN potrafi wynajdywać grupy o różnej gęstości. Działanie algorytmu opiera się na uporządkowaniu punktów w taki sposób, że sąsiednie punkty są blisko siebie w przestrzeni poszukiwań. Przy okazji dla każdego punktu przechowywane są odpowiednie wskaźniki, od których zależy przypisanie do poszczególnych klastrów.

Podczas wybierania algorytmów dążono do selekcji zbioru metod różnorodnych pod względem typu przeprowadzanego grupowania.

Implementacje powyższych algorytmów zaczerpnięto z modułu *sklearn.cluster*.

2.2. Wykorzystane miary jakości grupowania

Do oceny jakości przeprowadzonego grupowania przez poszczególne algorytmy, posłużono się następującymi miarami:

- miara **Rand** - **zewnętrzna miara** ewaluacji grupowania, która porównuje wyznaczone grupy z tymi wzorcowymi. Miara wyliczana jest według wzoru:

$$Rand = \frac{|\mathrm{TP}| + |\mathrm{TN}|}{\binom{n}{2}},$$

gdzie:

- ➢ *TP* - zbiór par obiektów, z których każda jest zawarta w pewnej grupie wzorcowej i w pewnej grupie wyznaczonej,
- ➢ *TN* - zbiór par obiektów, z których każda nie jest zawarta w żadnej grupie wzorcowej i w żadnej grupie odkrytej,
- ➢ *n* - liczba obiektów.

Miara Rand przyjmuje wartości od 0 do 1 - wyższa wartość wskazuje na lepsze grupowanie. Na poziomie implementacji wykorzystano funkcję *rand_score()* z modułu *sklearn.metrics*.

- miara **Silhouette** (***wskaźnik sylwetkowy***) - **wewnętrzna miara** ewaluacji grupowania, która wykorzystuje do oceny jedynie właściwości wyznaczonych grup. Miara wyliczana jest jako średnia wartość *S(i)* dla wszystkich punktów zbioru danych:

$$S(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}},$$

gdzie:

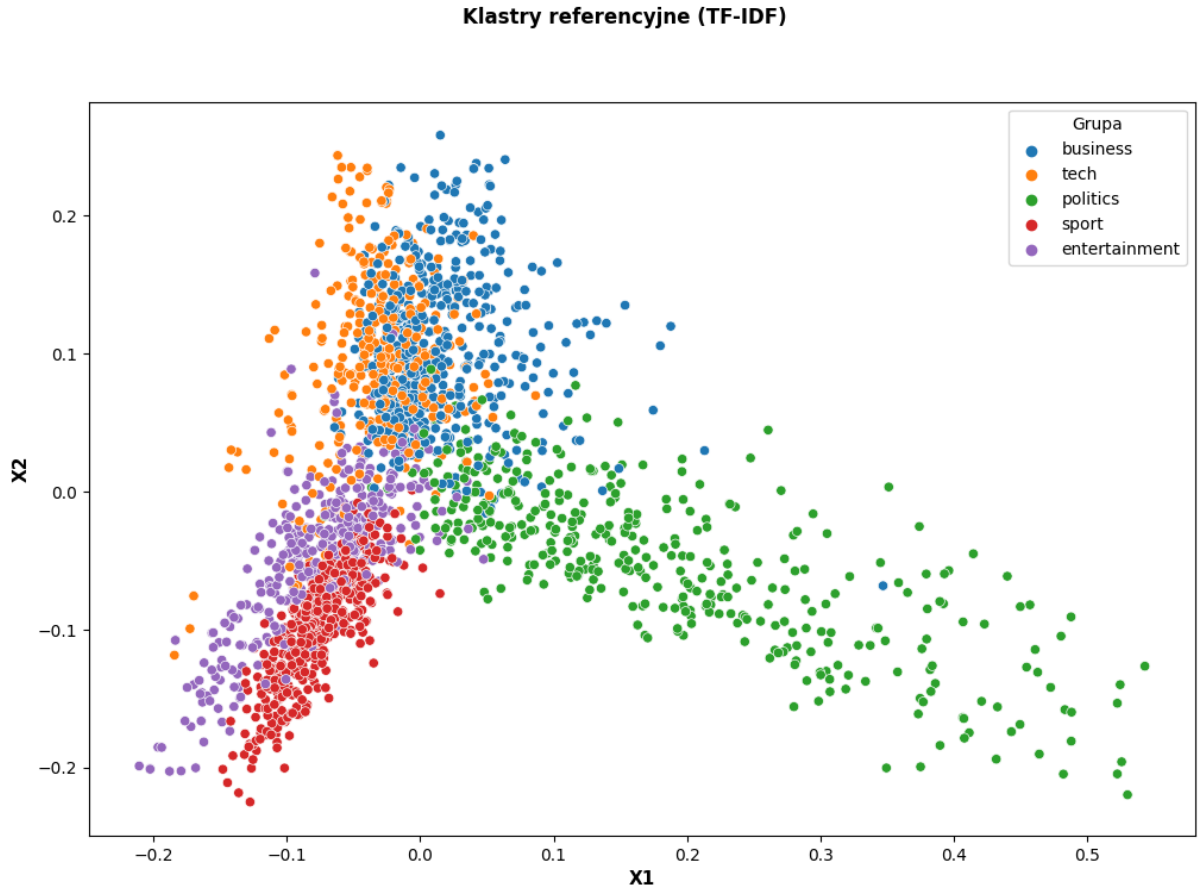
- ➢ *i* - indeks punktu zbioru danych,
- ➢ *a(i)* - średnia odległość od punktu *i* do innych punktów należących do tej samej grupy, co punkt *i*,
- ➢ *b(i)* - średnia odległość od punktu *i* do wszystkich innych punktów grupy, która jest najbliższą grupą względem grupy zawierającej punkt *i*.

Wskaźnik sylwetkowy przyjmuje wartości od -1 do 1, gdzie -1 oznacza słabą jakość grupowania, a wyższe wartości lepsze grupowanie (bardziej gęste i lepiej odseparowane od siebie klastry). Wartości bliskie 0 wskazuję na to, że grupy na siebie nachodzą.

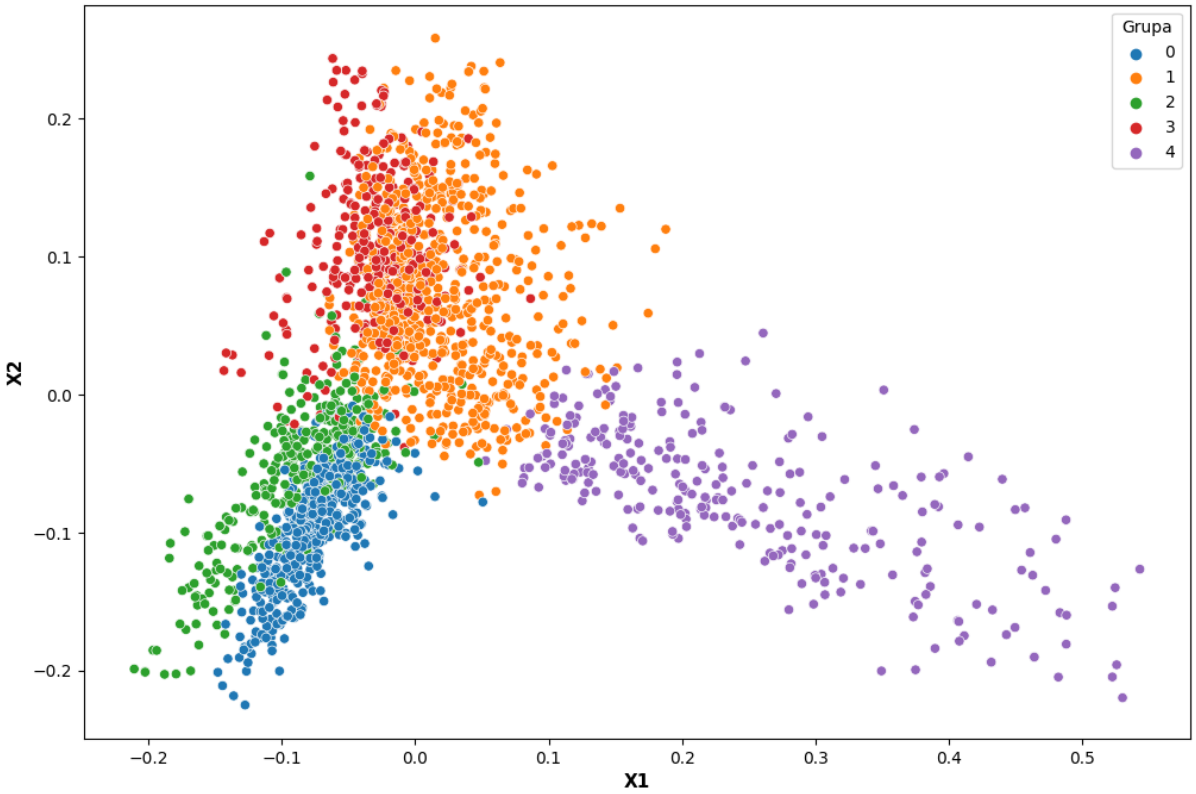
Na poziomie implementacji miary Silhouette wykorzystano funkcję *silhouette_score()* z modułu *sklearn.metrics*.

2.3. Porównanie wizualne algorytmów grupowania

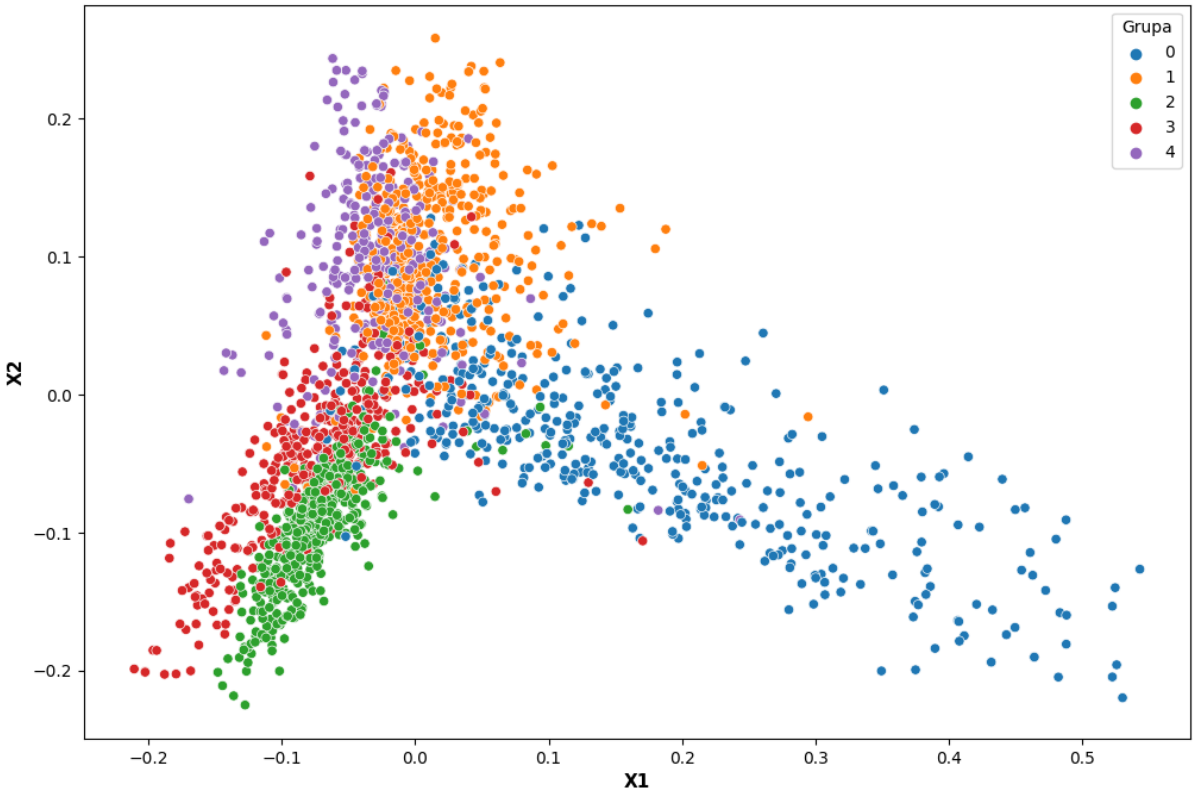
Poniżej przedstawiono na odpowiednich wykresach grupy odkryte przez algorytm hierarchicznego grupowania, k-średnich i OPTICS. Wyznaczone grupy artykułów porównano z referencyjnymi klastrami. Na pierwszy rzut oka można dostrzec wysoką jakość grupowania przeprowadzonego przez algorytm k-średnich i hierarchicznego grupowania, i to przy użyciu różnego rodzaju metod wektoryzacji tekstu (TD-IDF, Glove). Wyraźnie prezentuje się również wyjątkowo niska jakość grupowań przeprowadzonych przez algorytm OPTICS. Jakość ta wydaje się być nawet niższa niż jakość grupowań algorytmu NBC (chociaż ta też pozostawia wiele do życzenia). Już teraz można postawić hipotezę, że to struktura samych danych w przestrzeni przeszukiwań wpływa na niezadowalającą jakość grupowań gęstościowych. Hipoteza ta została zweryfikowana w kolejnej sekcji dokumentacji, która komentuje wartości wyliczonych metryk ewaluacji grupowania.



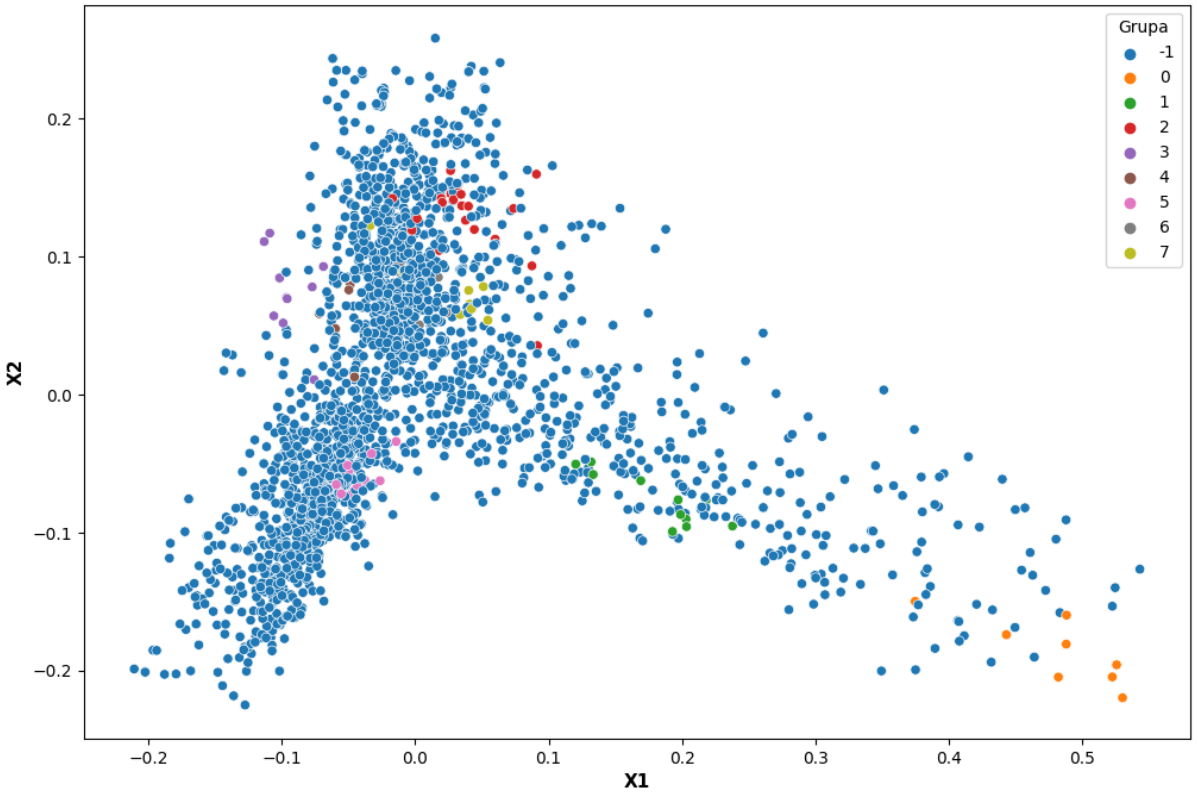
Klastry wyznaczone przez K-Means (TF-IDF)



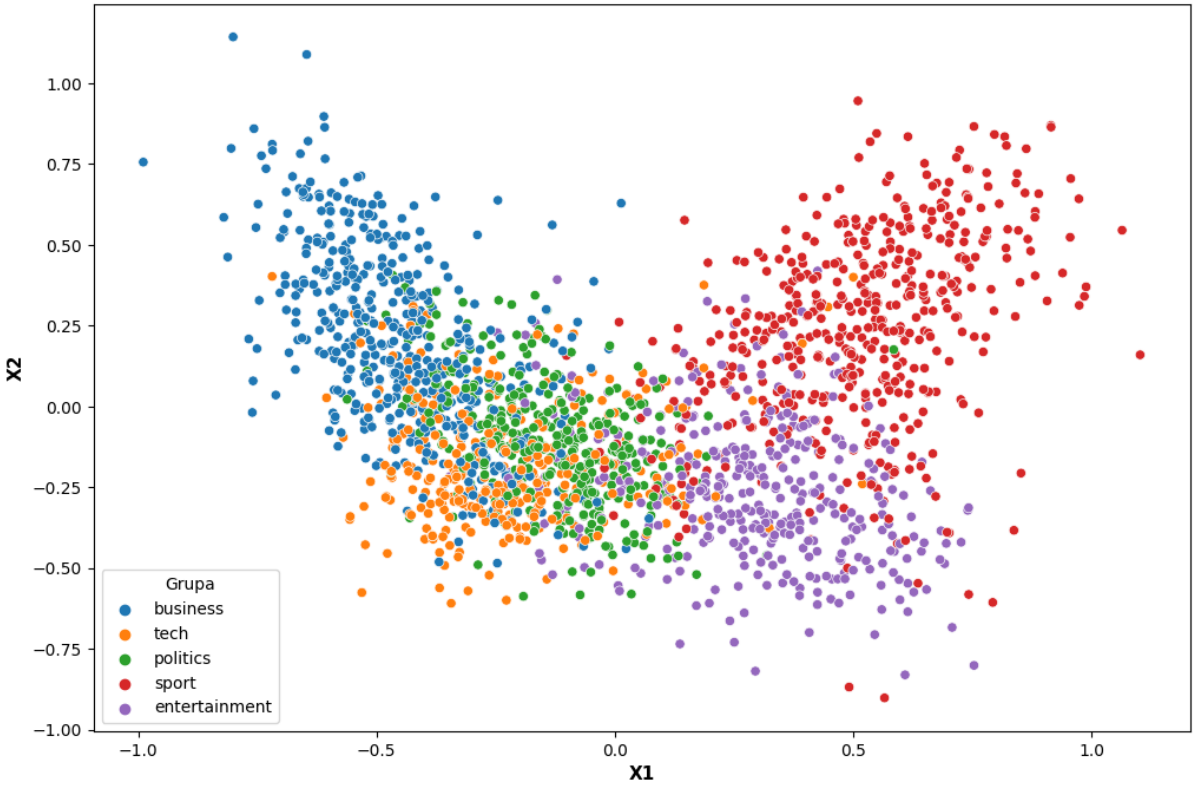
Klastry wyznaczone przez Agglomerative Clustering (TF-IDF)



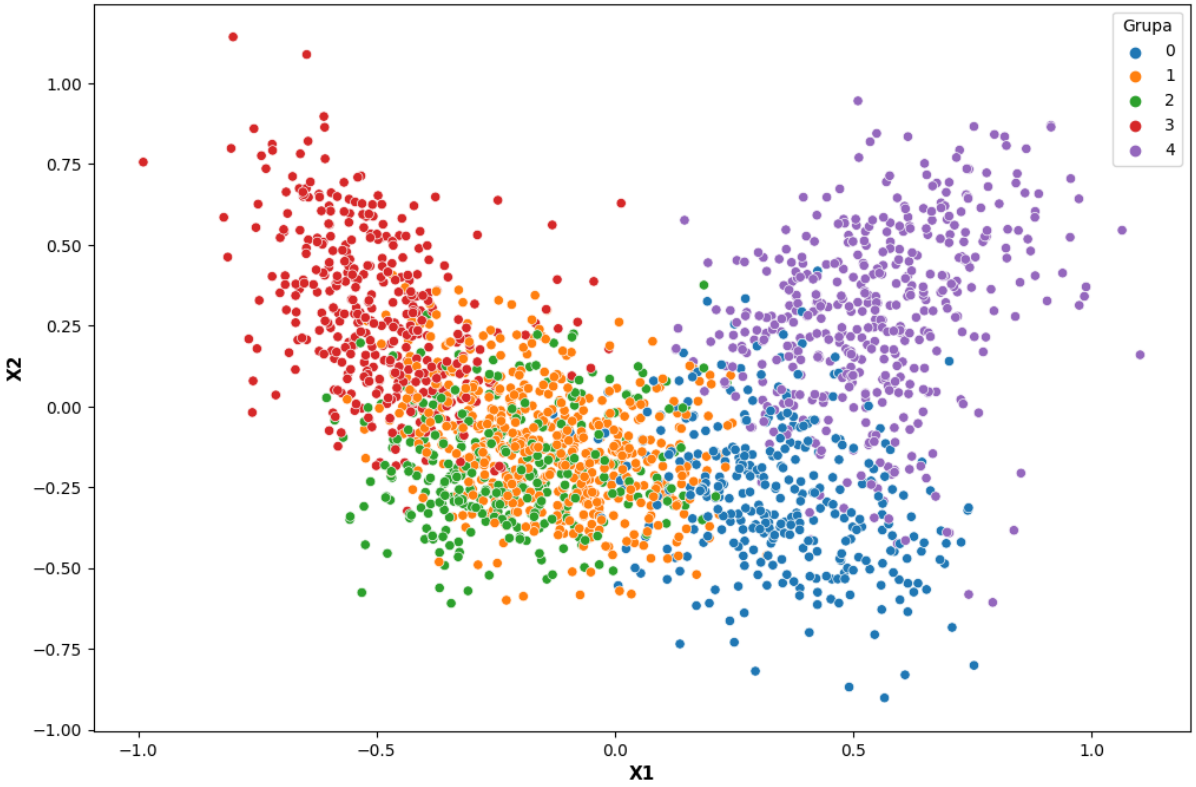
Klastry wyznaczone przez OPTICS (TF-IDF)



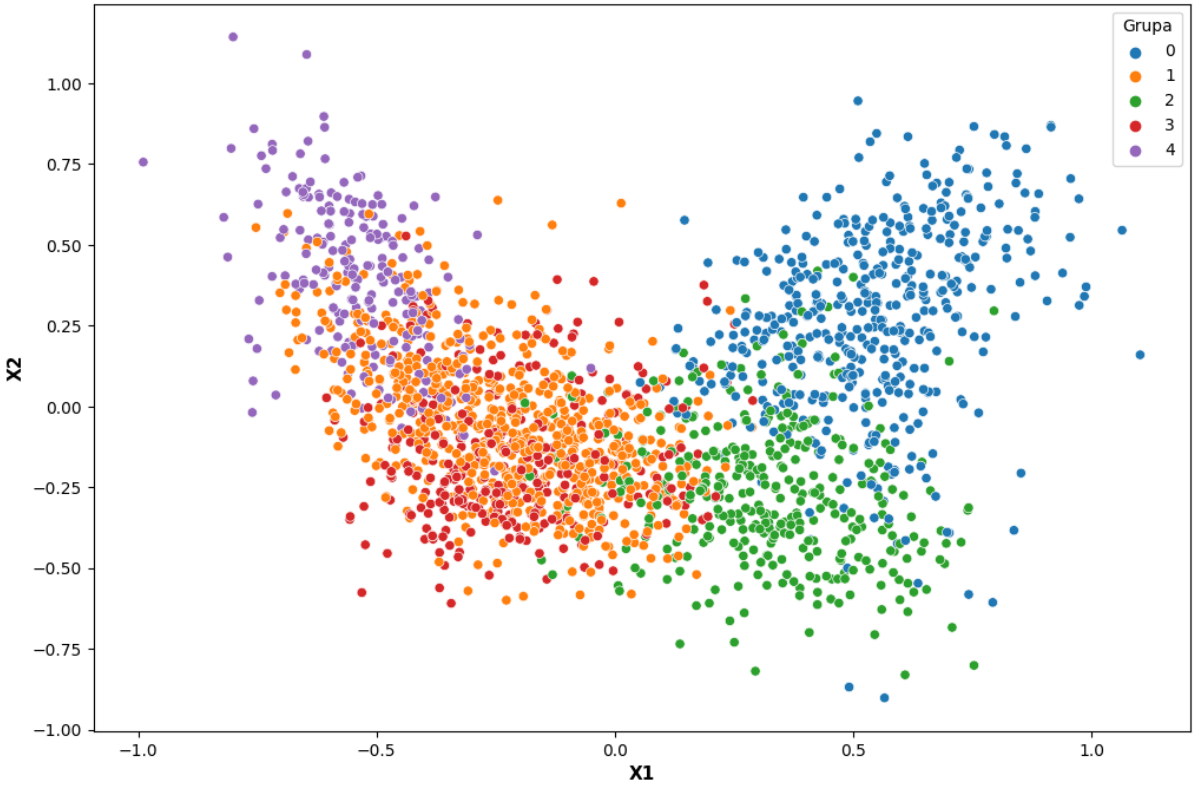
Klastry referencyjne (GloVe)

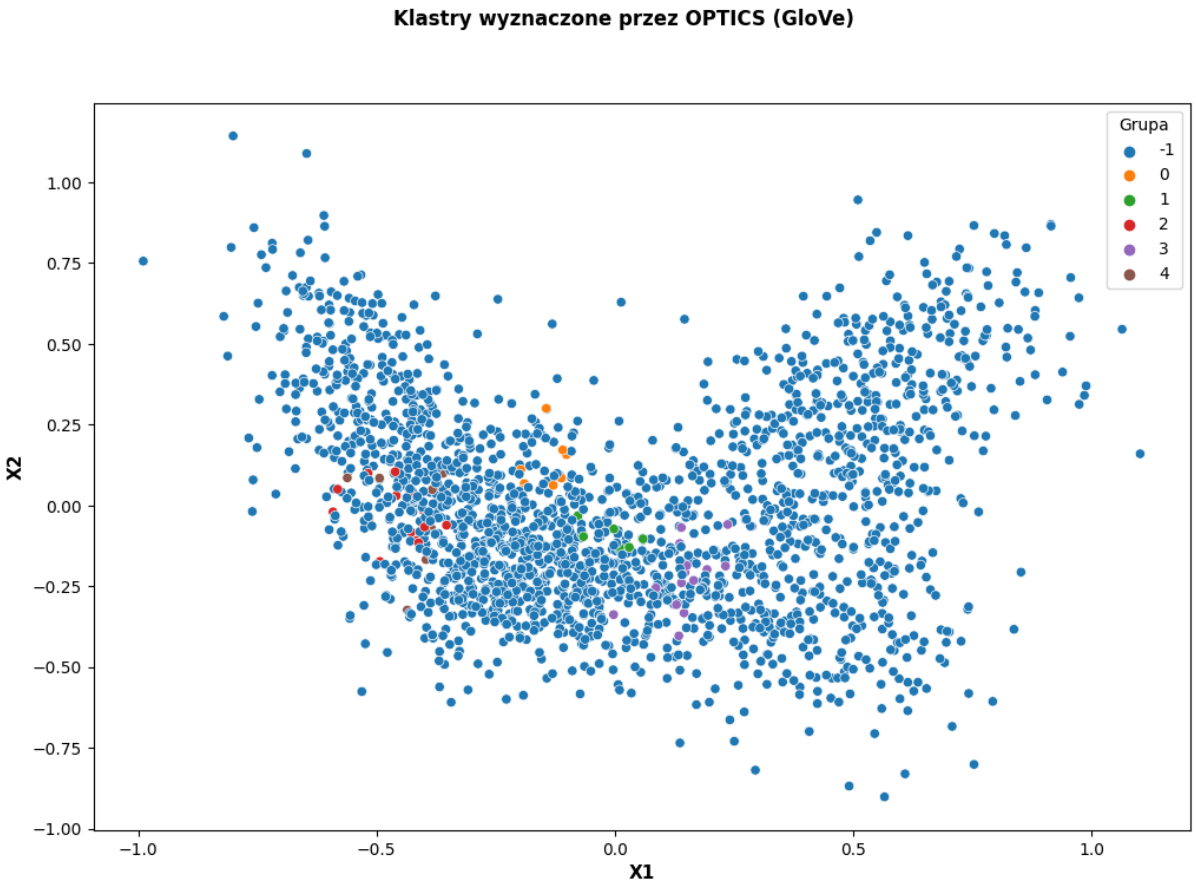


Klastry wyznaczone przez K-Means (GloVe)



Klastry wyznaczone przez Agglomerative Clustering (GloVe)





2.4. Analiza metryk ewaluacji grupowania

W poniższej tabelce przedstawiono wartości miar Rand i wskaźnika sylwetkowego dla poszczególnych grupowań przeprowadzonych przez badane w ramach projektu algorytmy:

Algorytm	$Rand \in [0, 1]$		$Silhouette \in [-1, 1]$	
	TF-IDF	***GloVe***	***TF-IDF***	***Glove***
NBC	**0,4**	**0,463**	**-0,008**	**-0,201**
k-średnich	**0,909**	**0,907**	**0,015**	**0,15**
hierarchiczny	**0,927**	**0,873**	**0,014**	**0,136**
OPTICS	**0,255**	**0,232**	**-0.008**	**-0,148**
	Test 1	***Test 2***	***Test 1***	***Test 2***
NBC	**0,833**	**0,965**	**-0,017**	**0,641**

Dla wszystkich metod grupowania zbioru artykułów wartości wskaźnika sylwetkowego oscylowały wokół zera. Wynika z tego, że wyznaczane grupy w dużym stopniu nakładały się na siebie. Biorąc pod uwagę też wysoką jakość grupowania algorytmu hierarchicznego czy

k-średnich (miara Rand), można przypuszczać, że same klastry referencyjne się ze sobą mocno pokrywają. Prawdopodobnie właśnie słaba separowalność pomiędzy grupami wpłynęła w dużej mierze na niską jakość grupowań przeprowadzonych przez algorytmy grupowania gęstościowego (NBC i OPTICS). Z drugiej strony, ze zbiorem testowym 1 algorytm NBC poradził sobie całkiem przyzwoicie, pomimo równie niskiego wskaźnika sylwetkowego. Zachodzi więc podejrzenie, że wpływ na grupowanie gęstościowe ma również wymiar samej przestrzeni grupowania (dla zbiorów testowych przestrzeń miała tylko 2 wymiary).

2.5. Podsumowanie II etapu

Analizy różnego typu algorytmów grupowania potwierdziły tezę, że niska jakość działania algorytmu NBC na danych eksperymentalnych jest ściśle związana ze strukturą zbioru artykułów BBC. Najprawdopodobniej niska separowalność referencyjnych grup wpłynęła negatywnie na efekty metod grupowania gęstościowego. Inne typy grupowania - hierarchicznego i iteracyjno-optymalizacyjnego - poradziły sobie z postawionym zadaniem w znacząco lepszym stopniu.

3. Bibliografia

[1] S. Zhou, Y. Zhao, J. Guan, i J. Huang, „A Neighborhood-Based Clustering Algorithm”, w Advances in Knowledge Discovery and Data Mining, Berlin, Heidelberg, 2005, s. 361–371. doi: 10.1007/11430919_43.

[2] M. Kryszkiewicz i P. Lasek, „A Neighborhood-Based Clustering by Means of the Triangle Inequality”, w Intelligent Data Engineering and Automated Learning – IDEAL 2010, Berlin, Heidelberg, 2010, s. 284–291. doi: 10.1007/978-3-642-15381-5_35.

[3] „Comparing different clustering algorithms on toy datasets”, scikit-learn. https://scikit-learn/stable/auto_examples/cluster/plot_cluster_comparison.html (dostęp 30 grudzień 2022).

[4] „GloVe: Global Vectors for Word Representation”. https://nlp.stanford.edu/projects/glove/ (dostęp 20 grudzień 2022).

[5] „Łukasz Jarosław Reszka / Algorithm NBC - MED 2022Z · GitLab”, GitLab. https://gitlab-stud.elka.pw.edu.pl/lreszka/algorithm-nbc-med-2022z (dostęp 20 grudzień 2022).

[6] „ML Resources - BBC Datasets”. http://mlg.ucd.ie/datasets/bbc.html (dostęp 17 grudzień 2022).

[7] S. Subedi, „NLP with Python: Text Clustering”, Sanjaya’s Blog, 12 maj 2019. https://sanjayasubedi.com.np/nlp/nlp-with-python-document-clustering/ (dostęp 17 grudzień 2022).

[8] „Porter Stemming Algorithm”. https://tartarus.org/martin/PorterStemmer/ (dostęp 18 grudzień 2022).

[9] S.-N. arvindpdmn, „Text Clustering”, Devopedia, 8 grudzień 2019. https://devopedia.org/text-clustering (dostęp 19 grudzień 2022).

[10] N. Kapur, „Text Preprocessing — NLP Basics”, Analytics Vidhya, 18 lipiec 2020. https://medium.com/analytics-vidhya/text-preprocessing-nlp-basics-430d54016048 (dostęp 17 grudzień 2022).

[11] D. Cam-Stei, „TF-IDF vs Word Embedding, a comparison and code tutorial", Medium, 19 luty 2019. https://medium.com/@dcameronsteinke/tf-idf-vs-word-embedding-a-comparison-and-code-tutorial-5ba341379ab0 (dostęp 17 grudzień 2022).

[12] M. Klimaszewski, „(TI-)NBC". 28 czerwiec 2021. Dostęp: 5 grudzień 2022. [Online]. Dostępne na: https://github.com/mklimasz/TI-NBC