Dokumentacja - algorytm NBC

Autor: Jakub Szczepanowski

1. Opis zadania

NBC (Neighborhood-Based Clustering) to algorytm grupowania gęstościowego. W porównaniu do algorytmu KMeans, w którym porównywane są odległości poszczególnych elementów z centroidami, grupowanie gęstościowe opiera się na otoczeniu każdego z punktów określając go - w zależności od wyznaczonego progu gęstości - jako rdzeniowy, jako nierdzeniowy, ale należący do danej grupy lub jako szum/element odstający/anomalia. Celem zadania jest zbadanie właściwości algorytmu NBC jak i jakości własnej implementacji w tym oceny wymagań potrzebnych do funkcjonowania algorytmu.

2. Opis implementacji

Projekt zrealizowano w języku Python z wykorzystaniem popularnych pakietów: Scikit Learn, NumPy, SciPy, Matplotlib, Pandas.

Struktura całego projektu składa się z kilku plików:

- main.py - główny plik potrzebny do wygodnego uruchomienia programu bez potrzeby bezpośredniego korzystania z API,
- config.json - plik konfiguracyjny zawierający wszystkie kluczowe parametry przy uruchamianiu programu,
- experiments.ipynb - plik notatnika Jupyter zawierający testy na oprogramowaniu,
- clustering.py - moduł główny rozwiązania, to w nim znajduje się implementacja algorytmu NBC,
- testing.py - moduł pomocniczy, zawiera funkcje wykorzystywane przy testowaniu algorytmu.

Szczegóły klasy NBC:

- rola estymatora (metoda fit) to obliczenie odległości pomiędzy obiektami,
- funkcję odległości oparto o uogólnienie metryki odległości zwanej odległością Minkowskiego,
- rola predyktora (metoda predict) to obliczenie wyników grupowania wykorzystując przy tym odległości obliczone za pomocą estymatora. Obliczenie tych wyników dokonuje się przez wyznaczenie k najbliższych sąsiadów każdego z punktów uwzględniając przy tym możliwość występowania punktów granicznych o takiej samej odległości oraz odwrotnych k sąsiadów. Na podstawie liczby poszczególnych sąsiadów obliczany jest współczynnik NDF (współczynnik gęstości). Jeżeli jest on większy lub równy 1 punkt uznawany jest za rdzeniowy. Sam mechanizm przypisywania obiektów do klastrów to przetłumaczony i dostosowany pod język Python pseudokod zamieszczony w artykule naukowym dotyczącym NBC [1],

- zarówno obliczanie odległości jak i znajdowanie k najbliższych sąsiadów zrealizowano na dwa różne sposoby: pierwszy zwany „normalnym” wykorzystuje zarówno struktury danych jak i metody dostępne w standardowej bibliotece Python’a, drugi „zoptymalizowany“ wykorzystuje metodę do optymalnego obliczania odległości z pakietu SciPy zaimplementowaną w języku C co gwarantuje szybkość całej operacji oraz funkcje pakietu NumPy do sprawniejszego obliczenia sąsiadów,
- szum oznaczono liczbą równą -1.

3. Opis przeprowadzonych eksperymentów

W przeprowadzonych eksperymentach skupiono się na badaniu zarówno użyteczności samego algorytmu dla zadań klasyfikacji jak i samej implementacji. W tym celu zastosowano kilka zbiorów oraz metryk ewaluacji.

Zbiory danych:

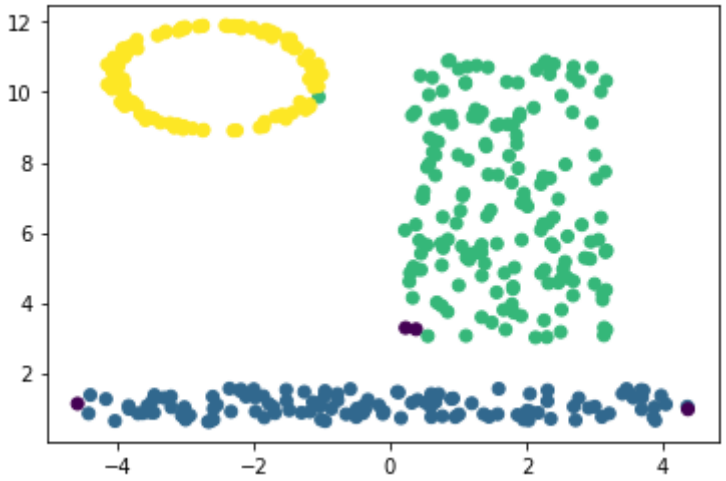
1. 3MC - sztuczny zbiór numeryczny - do testowania skuteczności NBC w konfiguracjach różnego kształtu i gęstości,
2. dense-disk-3000 - sztuczny zbiór numeryczny - „trudniejszy” zbiór do analizy uzyskiwanych wyników metryk w zależności od parametru k,
3. titanic - rzeczywisty zbiór mieszany - do testowania skuteczności algorytmu w rzeczywistych problemach klasyfikacyjnych oraz wpływu danych kategorycznych na wyniki,
4. vehicle - rzeczywisty zbiór numeryczny - do testowania skuteczności problemów klasyfikacyjnych na danych numerycznych,
5. USCensus1990 - rzeczywisty zbiór kategoryczny - do testowania skuteczności problemów klasyfikacyjnych na danych kategorycznych oraz przez rozmiary i wymiarowość skalowania ze względu na ilość elementów, ilość cech oraz zastosowaną metodą.

Metryki:

1. Rand - metryka ewaluacji zewnętrznej (wykorzystującej wiedzę ekspercką). W skali od 0 do 1 określa jakość grupowania na podstawie zewnętrznych klas, gdzie wynik równy 1 oznacza idealne grupowanie,
2. Davies-Bouldin - metryka ewaluacji wewnętrznej (nie wykorzystuje zewnętrznej wiedzy). Posiada wartości od 0 do nieskończoności, gdzie 0 oznacza idealną klasteryzację. Określa średnie podobieństwo klastrów jako stosunek odległości wewnątrz klastrów do odległości między klastrami.
3. Silhouette - metryka ewaluacji wewnętrznej (nie wykorzystuje zewnętrznej wiedzy). Posiada wartości między -1 a 1, gdzie 1 oznacza dobrą separowalność klastrów, 0 nakładające się klastry oraz liczby ujemne określają, że pewne próbki mogły zostać źle pogrupowane, ponieważ są bardziej podobne do innej grupy.

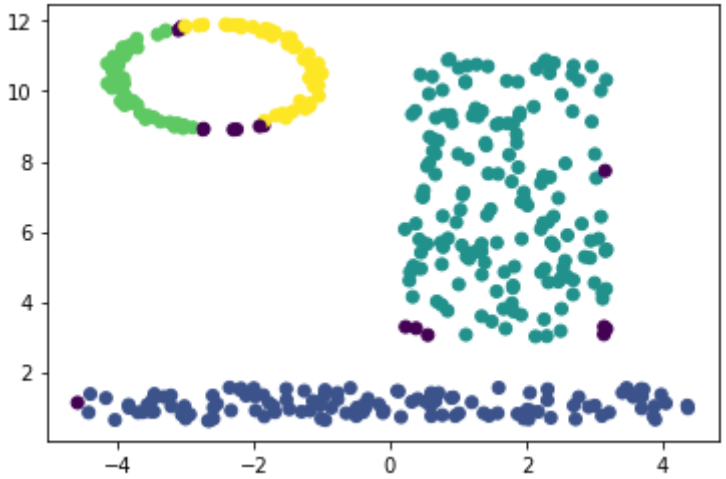
3.1. Wpływ skalowania cech na uzyskiwane wyniki

Pierwszym aspektem, na którym skupiono się w eksperymentach jest wpływ skalowania na wyniki grupowania. Wybór o stosowaniu lub nie skalowania cech mogło pozytywnie lub negatywnie wpłynąć na dalsze wyniki, dlatego była to pierwsza podjęta inicjatywa. Wpływ ten zbadano na zbiorze danych 3MC zawierającym skupiska danych o różnych kształtach i gęstościach.



Rys 1. Wyniki grupowania bez zastosowania skalowania cech

Wizualnie są one jak najbardziej akceptowalne. Wskazywał na to również wskaźnik Rand.



Rys. 2. Wyniki grupowania z zastosowaniem standaryzacji cech

Jak widać na powyższym rysunku skalowanie źle wpłynęło na wyniki grupowania głównie ze względu na skupisko danych eliptycznych. Użyto również standaryzacji z użyciem odchylenia przeciętnego, lecz wynik był tożsamy. Dodatkowo zaobserwowano, że po zeskalowaniu najlepsze wyniki uzyskano dla wartości parametru k o połowę niższym co pozytywnie wpływa na szybkości predykcji.

Wnioski są zatem takie, że skalowanie cech negatywnie wpływa na jakość grupowania.

3.2. Badanie skuteczności algorytmu ze względu na różne stopnie metryki odległości

Celem tej części badań było sprawdzenie, czy algorytm może uzyskać lepsze wyniki dla wyższych stopni metryki odległości.

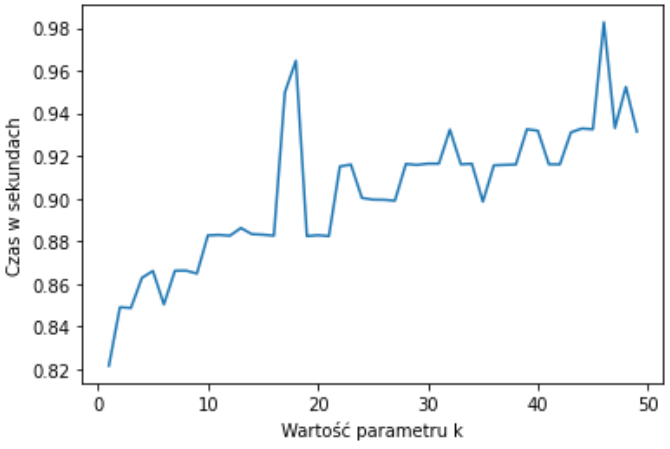
Stopień odległości	Wartość parametru k	Wartość metryki Rand
1 (Manhattan)	19	0.895
2 (Euklidesowa)	32	0.976
3	31	0.980
4	30	0.974
5	30	0.978

Tabela 1. Wyniki badania wpływu stopnia metryki odległości na jakość wyników

Jak widać na powyższej tabeli najlepszy wynik uzyskano dla stopnia równego 3. Tak jak różnica pomiędzy odległością Manhattan a euklidesową jest duża tak zaciera się ona w stopniach wyższych. Widać to również na przykładzie parametru k. Optymalna wartość k jest niemalże identyczna pomiędzy poszczególnymi wyższymi stopniami odległości.
Wniosek: można uzyskać lepsze wyniki grupowania przy wyższym stopniu metryki odległości, lecz te różnice są na tyle małe, że wystarczającą okazuje się odległość euklidesowa.

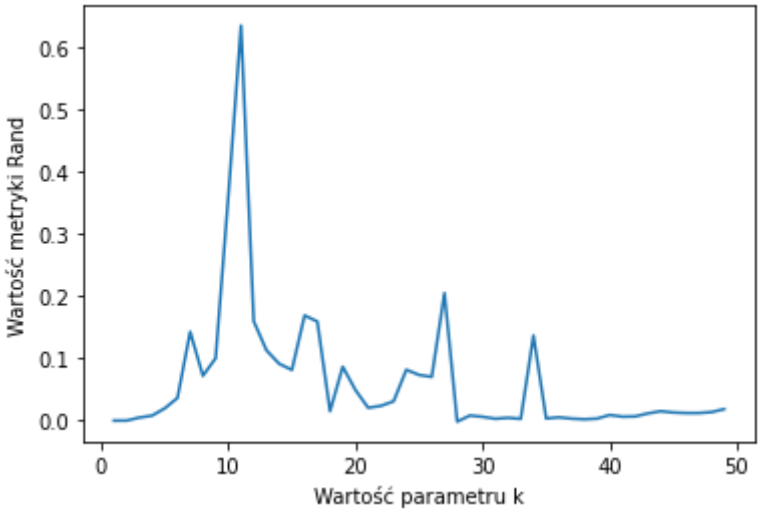
3.3. Badanie wpływu parametru k na czas oraz ostateczne wyniki grupowania

Ten eksperyment wykonano na zbiorze dense-disk-3000, który zawiera nachodzące na siebie eliptyczne grupy różniące się między sobą gęstością. Skupiono się tutaj na analizie wyników metryk ewaluacji oraz czasu wykonywania w zależności od parametru k.



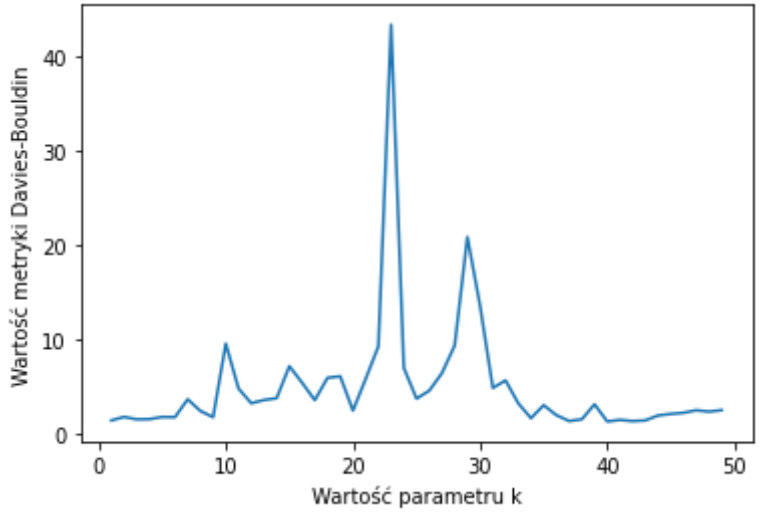
Rys. 3. Czas wykonywania w zależności od k

Wniosek: czas wykonywania w zależności od k - nie uwzględniając losowych skoków i szumów - rośnie logarytmicznie.



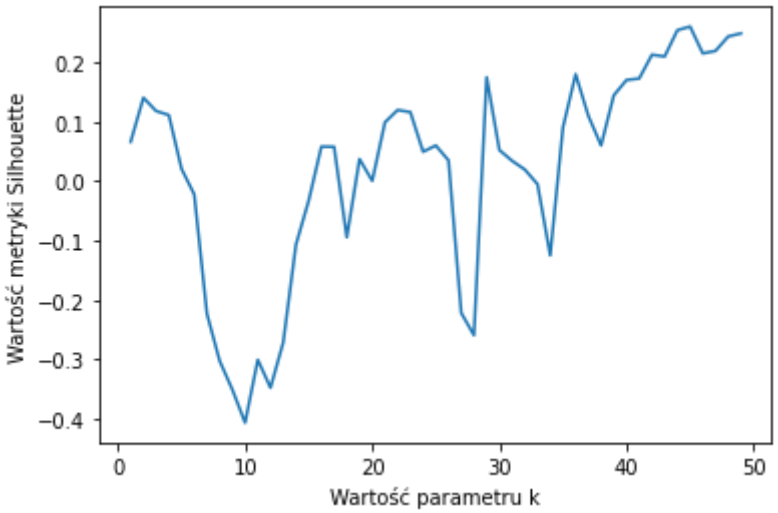
Rys. 4. Wartość metryki Rand w zależności od k

Wniosek: wartość współczynnika wybija się wyraźnie w jednej konkretnej wartości k co nie dobrze wróży, gdyż domaga się przeszukiwania przestrzeni parametru k w celu znalezienia satysfakcjonującego rozwiązania.



Rys. 5. Wartość metryki Davies-Bouldin w zależności od k

Wniosek: Tu jest podobnie, ale dla innej wartości k. W tej metryce nie wysokie, lecz niskie wartości są pożądane. Są one bliskie 0 dla albo bardzo małych albo bardzo dużych wielkości.



Rys. 6. Wartość metryki Silhouette w zależności od k

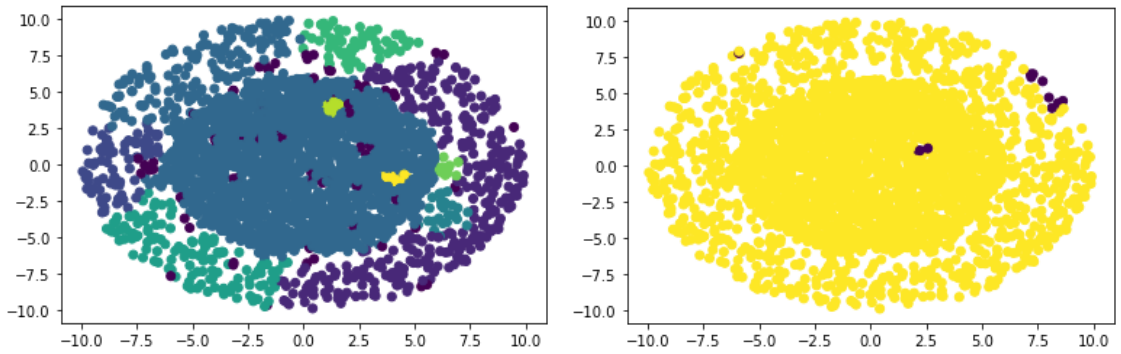
Wniosek: mocne wahania wskaźnika wskazują na duży wpływ parametru na ułożenie klastrów względem siebie. Przy wartości k takiej, gdzie dawała ona najlepsze wyniki w ewaluacji zewnętrznej metryka ta daje najniższe wyniki oznaczające ryzyko przypisania poszczególnych próbek do niewłaściwych klastrów, gdyż 0 wskazuje na nakładające się klastry. Może ona odzwierciedlać specyfikę zbioru - tego, że rzeczywiste klasy leżą blisko siebie. Współczynnik ten jest z pewnością zaniżany przez szum, który jest w tym przypadku interpretowany jako grupa.

Zbadano przy tym wpływ klasyfikowania szumu przez algorytm na wyniki metryk.

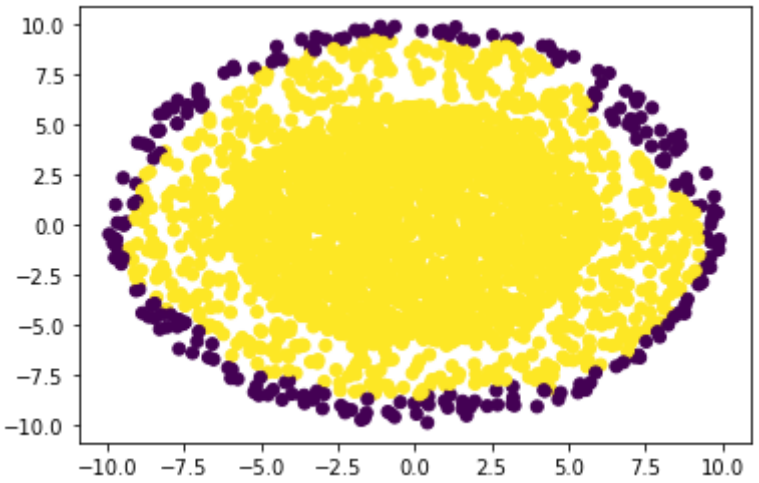
Nazwa metryki	Bez szumu	Z szumem
Rand	0.661	0.635
Davies-Bouldin	35.05	38.41
Silhouette	0.229	0.224

Tabela 2. Wpływ szumu na wyniki metryk

Wpływ szumu jest tutaj na tyle mały, że zdecydowano się nie usuwać go przy dalszych eksperymentach.
Dokonano również testów minimalizujących metrykę Davies-Bouldin oraz maksymalizujących Silhouette. Na podstawie wizualizacji wyników można wysnuć dodatkowe wnioski.



Rys. 7. Po lewej dążenie do maksymalizacji metryki Rand, a po prawej minimalizacja metryki Davies-Bouldin



Rys. 8. Wynik maksymalizacji metryki Silhouette

Wnioski: jak widać na obu rysunkach (Rys. 7 i Rys. 8) metryki wewnętrzne dążą do uznania całego tego skupiska jako jedną całość bo nie uwzględniają różnicy w gęstości dysków jako cechę odmienności.

3.4. Badanie wpływu danych kategorycznych na wyniki grupowania

Eksperyment wykonano przy użyciu zbioru danych titanic. Wycięto wiele cech zostawiając te najbardziej istotne w kontekście rozwiązania problemu przeżywalności podczas katastrofy statku. Wśród nich były zarówno dane numeryczne jak i kategoryczne. Uzyskany wynik Rand okazał się fatalnie niski bo wynoszący jedne 0.029. Po usunięciu danych kategorycznych nie zaobserwowano znaczącej poprawy (0.033). Zbadano również wpływ na rangowanie danych kategorycznych przekształcając je do postaci wektorów gorąco-jedynkowych. Nie zaobserwowano żadnej zmiany wyników.

Wnioski: Algorytm NBC nie jest wrażliwy zarówno na rangowanie danych kategorycznych jak i ich obecność w zbiorze. Mimo to nie nadaje się on do rozwiązywania problemów klasyfikacji.

3.5. Badanie jakości grupowania na danych numerycznych

Eksperyment wykonano na rzeczywistym zbiorze vehicle. Celem było ponowne zbadanie zdolności algorytmu do rozwiązywania problemów klasyfikacyjnych, tym razem z użyciem samych danych numerycznych. Pierwszym krokiem było wstępne przetwarzanie polegające na usunięciu silnie skorelowanych cech, a następnie grupowanie. Uzyskany wynik Rand (0.0007) potwierdza tylko przypuszczenia o nieskuteczności algorytmu w rozwiązywaniu problemów klasyfikacyjnych.

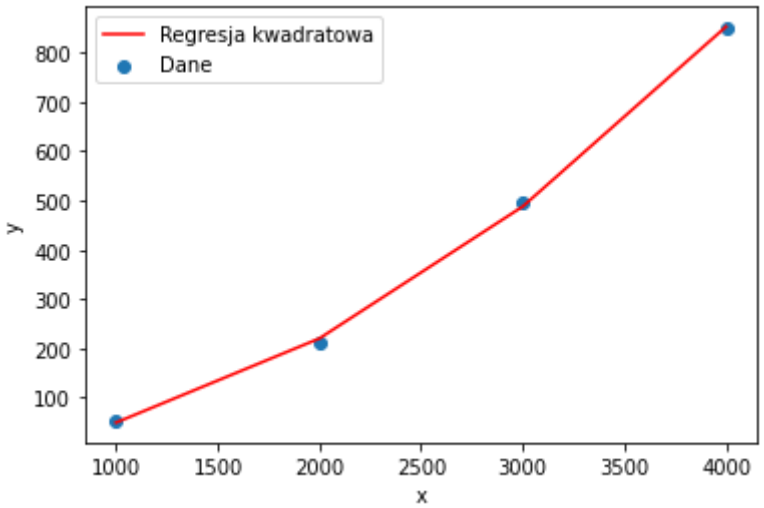
Wniosek: Wyniki klasyfikacji są złe głównie przez wykrywanie przez algorytm zmiennej liczby klas + szum. Dysproporcje pomiędzy liczbą rzeczywistych klas a predykowaną można redukować wielkością parametru k dopóki grupy są dobrze separowalne (tak jak w przypadku przedstawionych zbiorów sztucznych). Techniki wstępnego przetwarzania nieuwzględniające tego faktu nic nie dadzą, gdyż wewnętrzną specyfiką algorytmu jest wykrywanie gęstych skupisk danych bez zważania na to, że pewne klasy mogą podzielać między sobą cechy podobieństwa.

3.6. Badanie skalowalności algorytmu ze względu na wielkość danych wejściowych jak i ich wymiarowość

Eksperyment ten bada złożoność obliczeniową - specyfikę algorytmu, ale również metody implementacyjne. Wykonano to przy użyciu dużego zbioru kategorycznego USCensus1990.

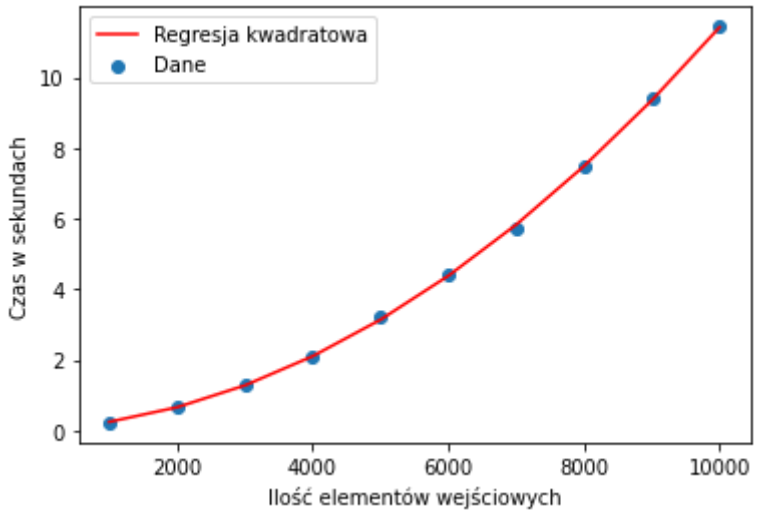
3.6.1. Skalowalność na liczbę elementów

Testy wykonano na ciągu liczby danych wejściowych od tysiąca do czterech tysięcy z krokiem tysiąc dla metody „normalnej”, a dla metody zoptymalizowanej było to od tysiąca do dziesięciu tysięcy również z krokiem tysiąc.



Rys. 9. Czas w sekundach w funkcji liczby danych wejściowych dla metody normalnej

Jak widać na rysunku wykres regresji kwadratowej idealnie nakłada się na wyniki pomiarowe.



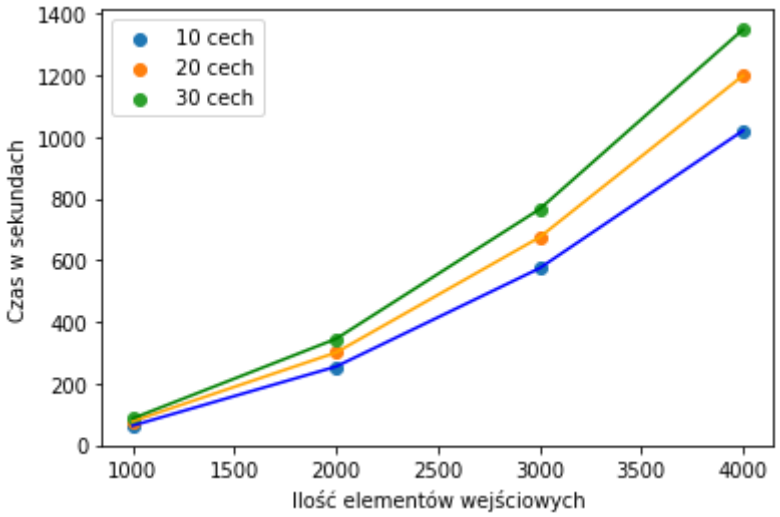
Rys. 10. Czas w sekundach w funkcji liczby danych wejściowych dla metody zoptymalizowanej

Na rysunku 10. jeszcze lepiej widać kwadratowy trend.

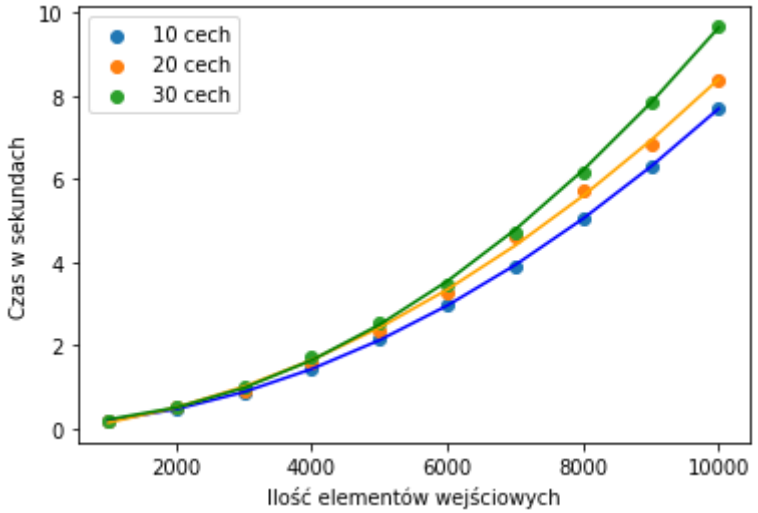
Wnioski: Wykorzystanie algorytmu NBC wiąże się z obliczeniem odległości pomiędzy obiektami (każdy z każdym) co bardzo mocno wpływa na czas obliczeń. Złożoność czasową w prosty sposób można wyprowadzić analitycznie co w tym przypadku potwierdzają tylko dane empiryczne. Złożoność pamięciowa w tym przypadku również będzie rosła kwadratowo co uniemożliwia przetwarzanie większych zbiorów danych. Bolączką w korzystaniu z NBC jest nie tylko czas, ale również zasoby pamięciowe.

3.6.2. Skalowalność na liczbę cech

Własność tę zbadano na trzech różnych ilościach cech: 10, 20, 30.



Rys. 11. Czas w sekundach w funkcji liczby danych wejściowych z różną wymiarowością dla metody normalnej



Rys. 12. Czas w sekundach w funkcji liczby danych wejściowych z różną wymiarowością dla metody zoptymalizowanej

Na obu wykresach widać, że czas przetwarzania zwiększa się nieznacznie w zależności od liczby cech.

Wnioski: porównując metodę normalną ze zoptymalizowaną widać wyraźnie, że Python jest wyraźnie wolniejszy od wersji napisanej w języku C. Mimo czasowego przeskoku wynikającego z zastosowanej technologii złożoność obliczeniowa dalej jest mało korzystna dla ogólnej użyteczności algorytmu.

4. Instrukcja użycia oprogramowania

Wymagania i zależności:

- interpreter 64-bitowy Python'a w wersji 3.9.6 lub wyższej,
- biblioteki, których specyfikacja zawarta jest w pliku requirements.txt,
- pliki programu: main.py (moduł główny), config.json (plik konfiguracyjny), clustering.py (moduł implementacyjny NBC).

Po zainstalowaniu interpretera w katalogu zawierającym plik główny programu należy wykonać komendę:

```
pip install -r requirements.txt
```

Następnie, można zmodyfikować plik konfiguracyjny, gdzie:

- dataset - to ścieżka do zbioru danych,
- format - to format w jakim znajduje się zbiór. Obsługiwane są formaty: „csv" i „arff",
- method - to metoda obliczania odległości oraz k najbliższych sąsiadów: dostępne metody to „normal" oraz „optimized". Zaleca się stosowanie metody zoptymalizowanej,

- k - pole numeryczne określające wielkość parametru k,
- plot - pole o wartości logicznej określające, czy użytkownik chce uzyskać wykres wizualizujący wynik grupowania,
- target_feature - pole określające nazwę atrybutu w zbiorze uważaną za atrybut docelowy (w zadaniach klasyfikacyjnych zawierający etykiety klas).

Po wprowadzeniu wymaganych pól oraz wykonania powyższej komendy można uruchomić program za pomocą komendy:

```
python main.py
```

Specyfikacja danych wejściowych oraz wyjściowych:
Obsługiwane przez program są zbiory tabelaryczne w formatach CSV i ARFF. Nie może on zawierać pól pustych! Może zawierać dane kategoryczne, które zostaną przekształcone na liczby całkowite. Obliczone przez program przypisania do grup są zapisywane do pliku output.npy w formacie NPY. Jest to format pochodzący z pakietu NumPy. Aby załadować tablicę wyników grupowania należy zaimportować moduł NumPy oraz użyć metody load.

```
import numpy as np
wyniki_grupowania = np.load('output.npy')
```

Oprócz tego program w terminalu wyświetli wyniki metryk ewaluacji w zależności od wybranych parametrów. Jeżeli użytkownik wybrał opcję „plot" wyświetlony zostanie wykres z wynikami grupowania. Jeżeli zbiór danych nie był dwuwymiarowy zostanie on sprowadzony do tej liczby wymiarów za pomocą techniki t-SNE - techniki redukcji wymiarowości stosowanej przy wizualizacji wykorzystującej rozkład t-studenta.

5. Wnioski

Dużą zaletą algorytmu NBC jest to, że posiada on tylko jeden parametr (k) co znacznie zmniejsza liczbę kombinacji potrzebną do uzyskania satysfakcjonującego rozwiązania. Czas ten nie tylko rośnie na skutek wielkości parametru, ale przede wszystkim na skutek kwadratowej złożoności obliczeniowej. Złożoności tej nie da się zlikwidować, gdyż pochodzi ona z wewnętrznej specyfiki algorytmu. Można ją jedynie zredukować stosując techniki optymalizacji sprzętowej/technologicznej lub zastosować jakąś heurystykę nie dającą w pełni optymalnych wyników (zaledwie ich przybliżenie). Dla porównania algorytm KMeans, aby mógł uzyskać taką złożoność jak NBC czy DBSCAN, iloczyn ilości centroidów i liczby iteracji musiałby być równy ilości wszystkich elementów w zbiorze. Mimo to NBC jest szybszy niż DBSCAN jak zarzekają się twórcy algorytmu [1]. Dużą wadą jest również bezużyteczność NBC w rozwiązywaniu rzeczywistych problemów klasyfikacyjnych co jest związane również ze specyfiką algorytmu uwzględniającą jedynie ilość k sąsiadów w otoczeniu każdego z obiektów oraz na jej podstawie wyznaczoną miarą gęstości. Jeżeli skupiska danych stykają się, bez względu na różnice w gęstości, algorytm będzie dążył do

„zjednoczenia” ich w jeden klaster - to wskazują chociażby wyniki metryk ewaluacji wewnętrznej. Użyteczność NBC leży w analizie skupień dobrze separowalnych co może stanowić informację eksploracyjną, czy decyzyjną w procesie etykietowania. Nie jest on wrażliwy na samą obecność jak i rangowanie danych kategorycznych co jest jak najbardziej jego zaletą. Algorytm do tego sprawnie wykrywa obwoluty klastrów etykietując je jako szum co również może stanowić istotną informację dla analityków.

6. Literatura

[1] Shuigeng Zhou, Yue Zhao, Jihong Guan, Joshua Huang, “A Neighborhood-Based Clustering Algorithm”, China, 2005.