Praca domowa 4

Jan Kwiecień 320565

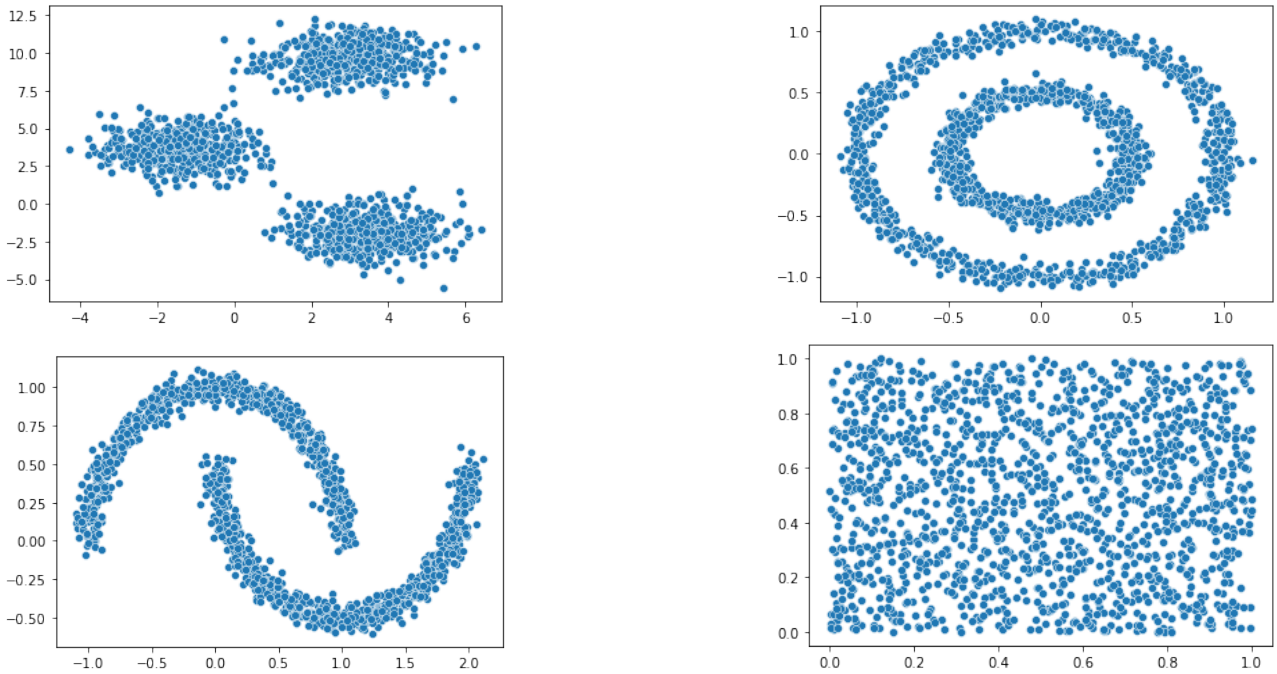
1 Wstęp

Celem pracy domowej numer 4 było przeprowadzenie trzech badań związanymi z metodami dla nienadzorowanego uczenia maszynowego (k - średnich, metody hierarchiczne i DBSCAN). W pierwszej części należało zbadać wpływ liczby klastrów na wyniki poszczególnych modeli. W drugiej trzeba było sprawdzić wpływ parametru *noise* na zbiory danych 2 oraz 3. Ostatnie zadanie to wygenerowanie wykresu pokazującego zależność całkowitej sumy odległości między punktami w klastrach od liczby klastrów. W celu rozwiązania powyższych zadań skorzystaliśmy z czterech sztucznie wygenerowanych zbiorów danych.

2 Część 1

2.1 Zbiory danych

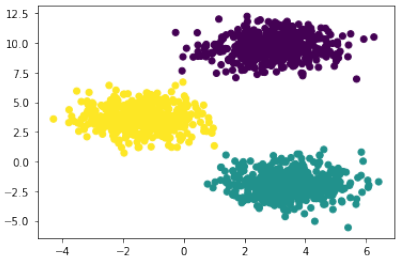
Zgodnie z poleceniem, wczytałem 4 zbiory danych, które prezentują się następująco:



Rysunek 1: Wizualizacje wczytanych zbiorów danych

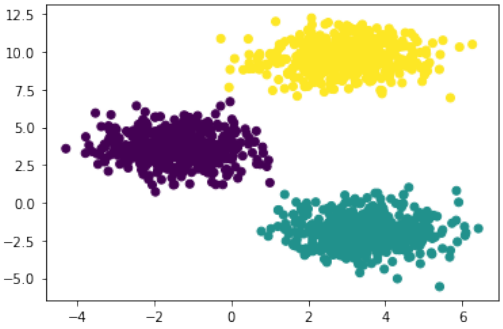
2.2 Zbiór 1

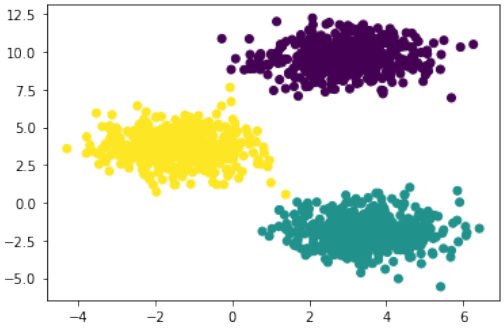
Zbiór 1 wraz z etykietami wygląda następująco:



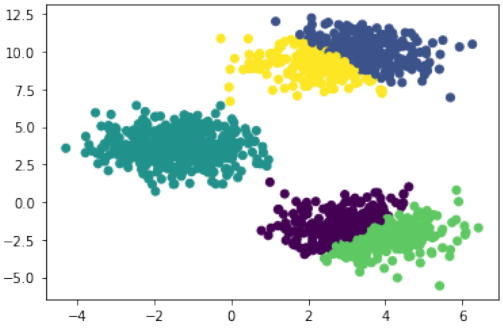
Rysunek 2: Zbiór 1 z etykietami.

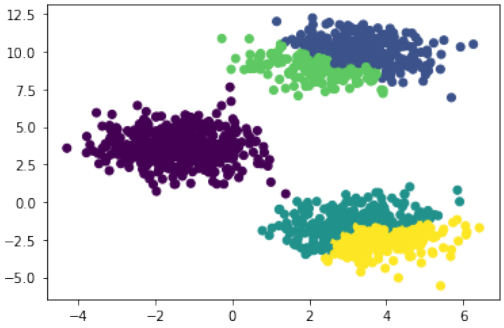
W tej części liczbę klastrów jaką badałem, to: 3, 5 oraz 10. Wyniki przedstawiają się następująco:



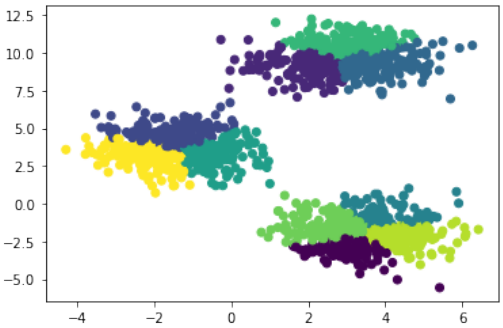


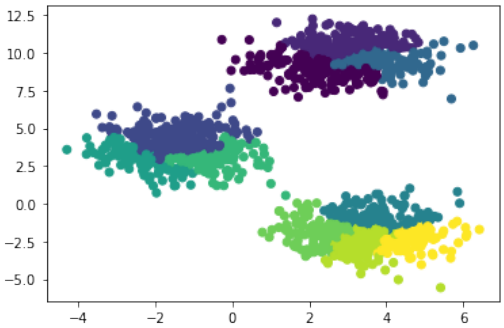
Rysunek 3: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 3, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 3.





Rysunek 4: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 5, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 5.



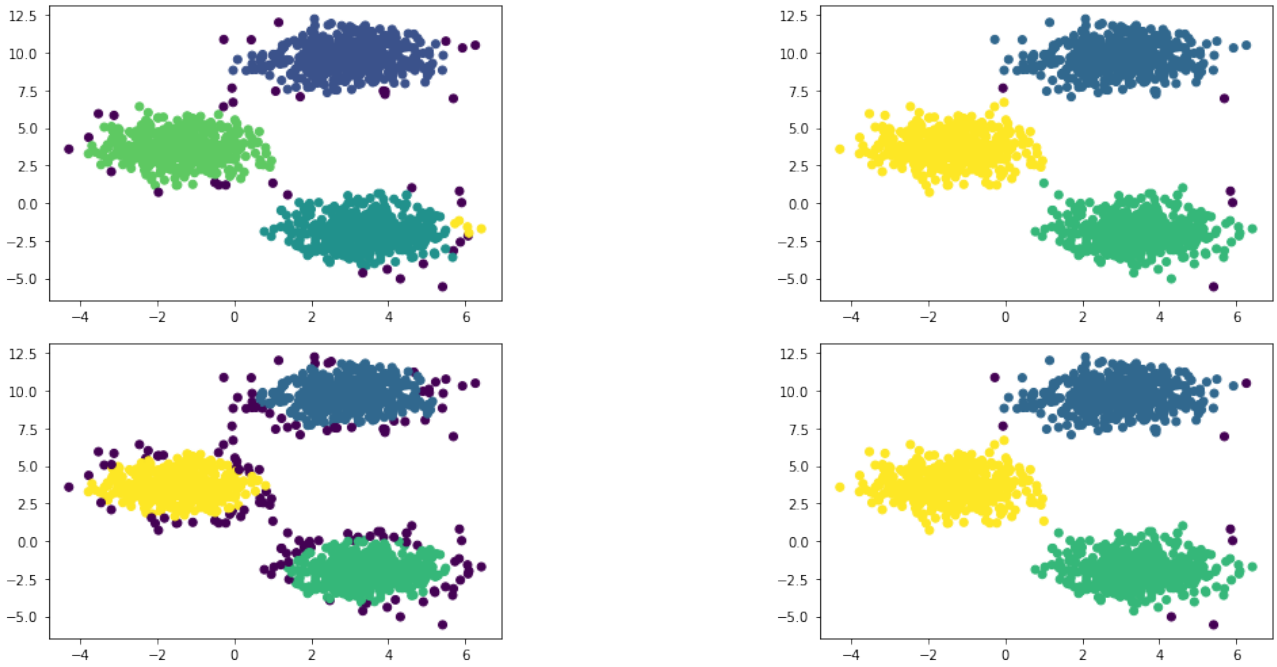


Rysunek 5: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 10, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 10.

Jak można zaobserwować przy użyciu liczby klastrów równej 3, model KMeans() poradził sobie bardzo dobrze, model AgglomerativeClustering() miał drobne pomyłki. Dla k = 5 modele dały podobne rozwiązania, a

gdy k = 10, całkowicie inaczej dokonały grupowania.

W przypadku DBSCAN nie mamy możliwości ustawienia liczby klastrów (jedynie parametr *eps* oraz *min_samples*). Z tego względu będę badał te wielkości w różnych zakresach. Wyniki dla zbioru danych 1 prezentują się w sposób następujący:

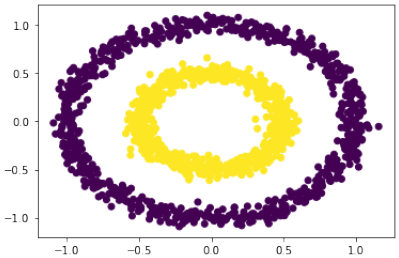


Rysunek 6: Dwa wykresy po lewej: DBSCAN z parametrami *eps* = 0.5 oraz *min_samples* = 5 i 15; dwa wykresy po prawej: DBSCAN z parametrami *eps* = 1 oraz *min_samples* = 5 i 15.

W przypadku, gdy *eps* = 0.5, metoda ta postrzega więcej punktów jako outliery. Dla *eps* = 1, poradziła sobie ona całkiem dobrze z grupowaniem.

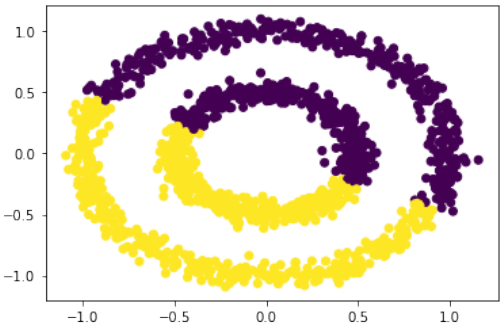
2.3 Zbiór 2

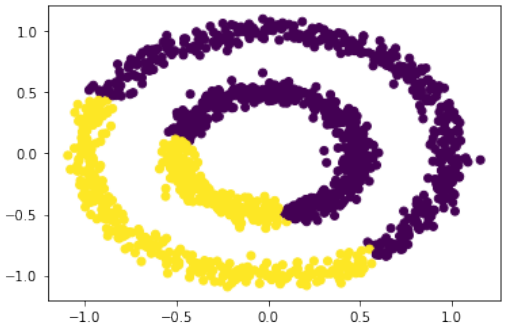
Zbiór 2 wraz z etykietami wygląda następująco:



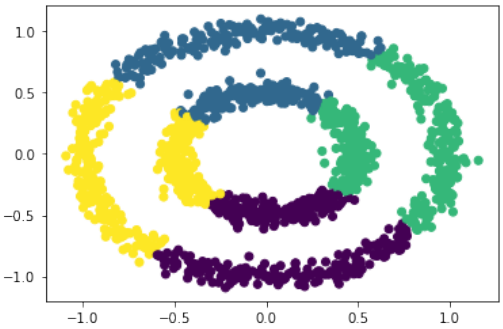
Rysunek 7: Zbiór 2 z etykietami.

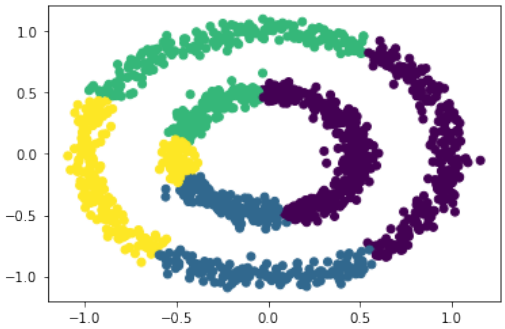
W tej części liczbę klastrów jaką badałem, to: 2, 4 oraz 8. Wyniki przedstawiają się następująco:



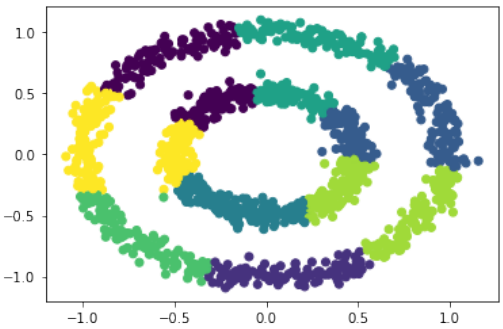


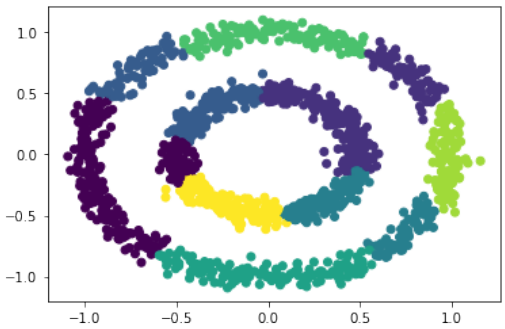
Rysunek 8: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 2, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 2.





Rysunek 9: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 4, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 4.

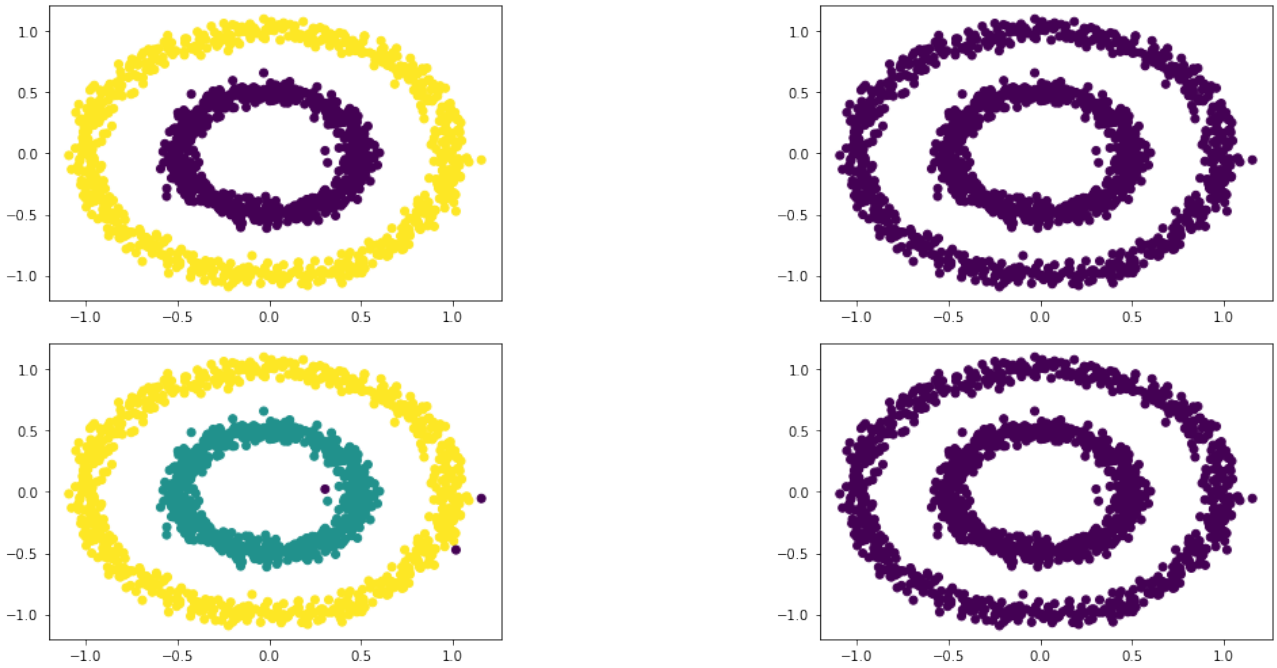




Rysunek 10: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 8, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 8.

Jak można zauważyć przy użyciu liczby klastrów równej 2, modele KMeans() i AgglomerativeClustering() dały podobne rezultaty, jednak różne od oryginalnego. Dla k = 4 i k = 8 oba modele trochę inaczej dokonały pogrupowań.

Wyniki DBSCAN wyglądają następująco:

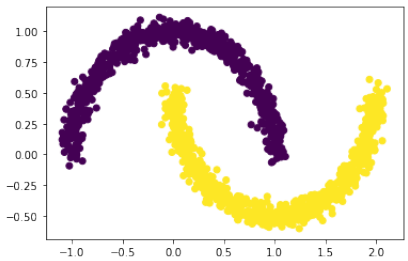


Rysunek 11: Dwa wykresy po lewej: DBSCAN z parametrami *eps* = 0.1 oraz *min_samples* = 5 oraz 15; dwa wykresy po prawej: DBSCAN z parametrami *eps* = 0.3 oraz *min_samples* = 5 i 15.

W przypadku, gdy *eps* = 0.1, metoda ta poradziła sobie bardzo dobrze w obydwu przypadkach (w drugim 3 outliery). Zmiana na *eps* = 0.3 spowodowała, że DBSCAN zakwalifikował wszystkie punkty do jednej grupy.

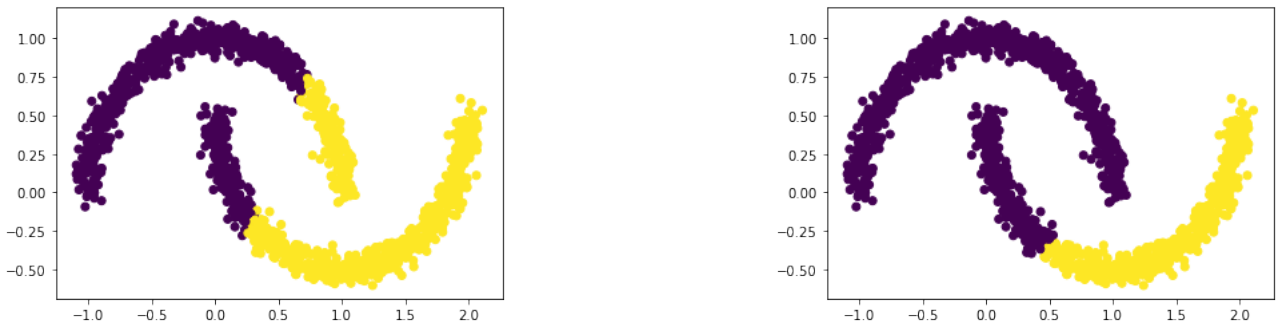
2.4 Zbiór 3

Zbiór 3 wraz z etykietami wygląda następująco:

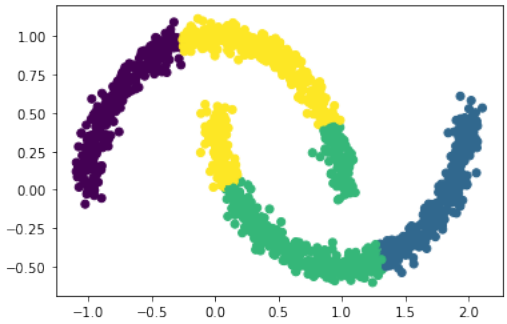
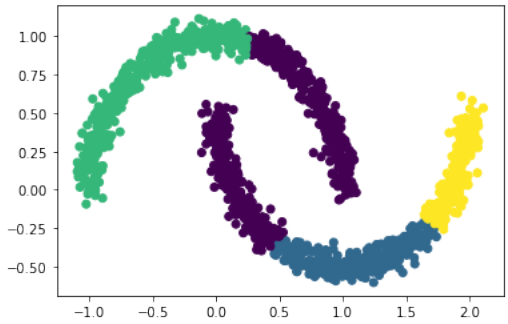


Rysunek 12: Zbiór 3 z etykietami.

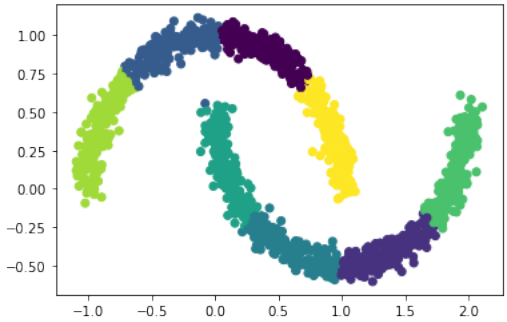
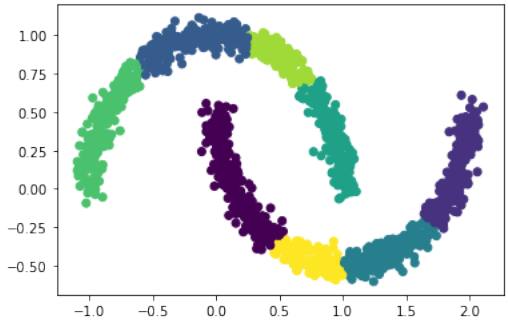
W tej części liczbę klastrów jaką badałem, to: 2, 4 oraz 8. Wyniki przedstawiają się następująco:



Rysunek 13: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 2, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 2.

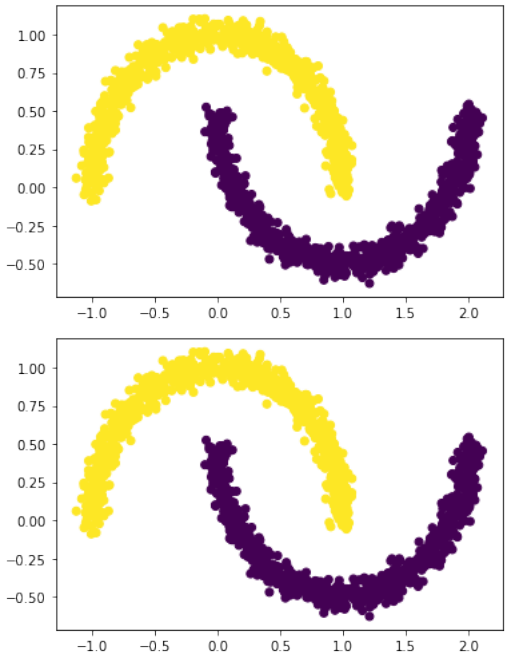
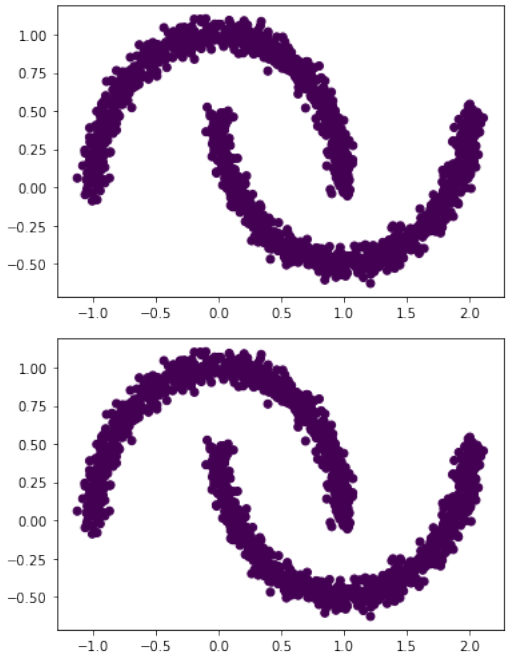
Rysunek 14: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 4, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 4.

Rysunek 15: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 8, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 8.

Po analizie można zobaczyć, że tym razem obydwa modele dały różne rezultaty dla k = 2, ponownie nieprawdziwe dla oryginalnych etykiet. Gdy k = 4 raz jeszcze obserwujemy różne wyniki, natomiast gdy k = 8, nie odnotowujemy dużych różnic.

Wyniki DBSCAN wyglądają następująco:

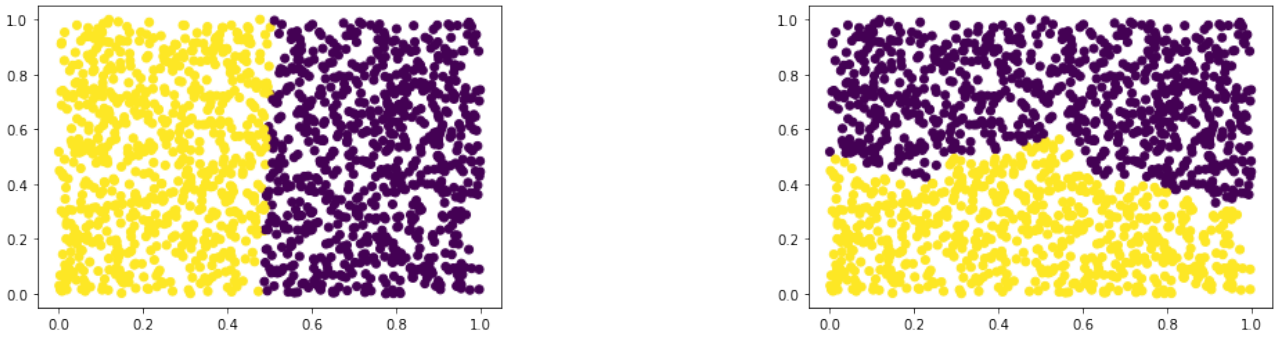
Rysunek 16: Dwa wykresy po lewej: DBSCAN z parametrami *eps* = 0.3 oraz *min_samples* = 5 oraz 15; dwa wykresy po prawej: DBSCAN z parametrami *eps* = 0.5 oraz *min_samples* = 5 oraz 15

Otrzymaliśmy podobne rezultaty co w poprzednim podpunkcie, dla *eps* = 0.3 klasteryzacja przebiegła po-

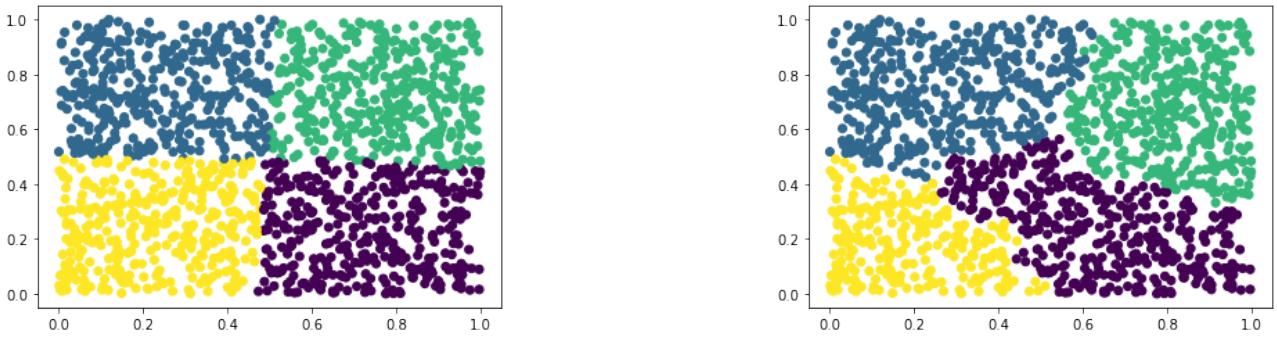
myślnie, dla $eps = 0.5$ wszystkie punkty zostały zakwalifikowane do jednej klasy.

2.5 Zbiór 4

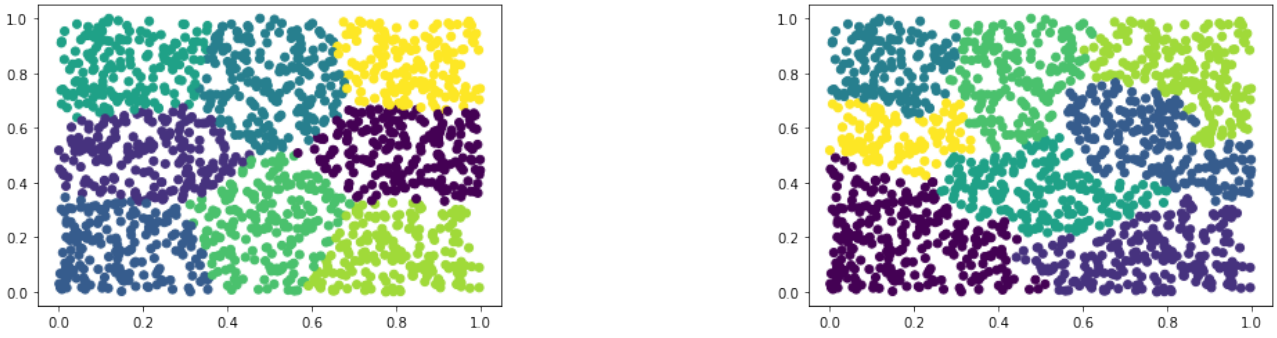
Ostatnim zbiorem są punkty wygenerowane losowo z rozkładu jednostajnego na [0,1] x [0,1]. Ponownie badam parametr k w zakresie: 2, 4 i 8. Wyniki prezentują się następująco:



Rysunek 17: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 2, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 2.



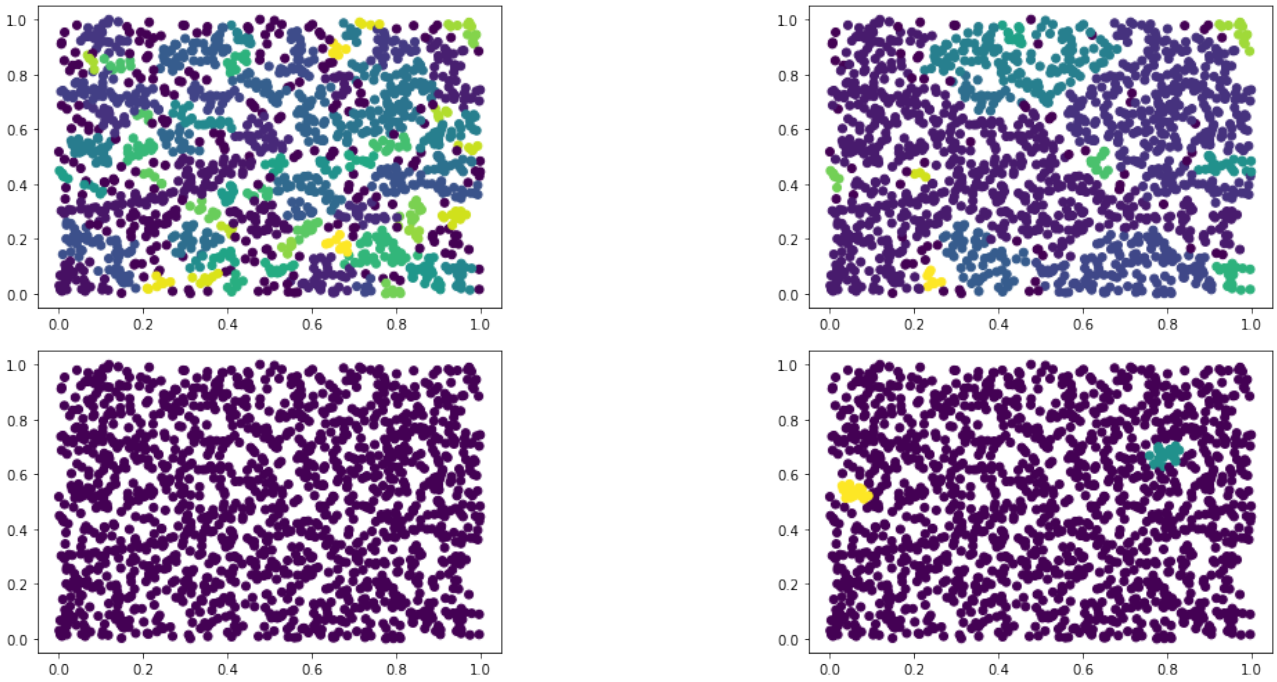
Rysunek 18: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 4, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 4.



Rysunek 19: Po lewej: klasteryzacja przy użyciu KMeans() z liczbą klastrów 8, po prawej: klasteryzacja przy użcyiu AgglomerativeClustering() z liczbą klastrów 8.

Dla każdego k metoda KMean() i AgglomerativeClustering() dała całkowicie odmienne rezultaty.

Wyniki DBSCAN wyglądają następująco:



Rysunek 20: Dwa wykresy po lewej: DBSCAN z parametrami *eps* = 0.03 oraz *min_samples* = 5 oraz 15; dwa wykresy po prawej: DBSCAN z parametrami *eps* = 0.035 oraz *min_samples* = 5 oraz 15

Tym razem dużą rolę odegrał również parametr *min_samples*. Gdy wynosił on 15 (dwa dolne rysunki), otrzymaliśmy jedną klasę (w drugim przypadku dwa dodatkowe małe klastry). Gdy *min_samples* = 5, podziały na grupy były losowe, w zależności od zagęszczenia punktów.

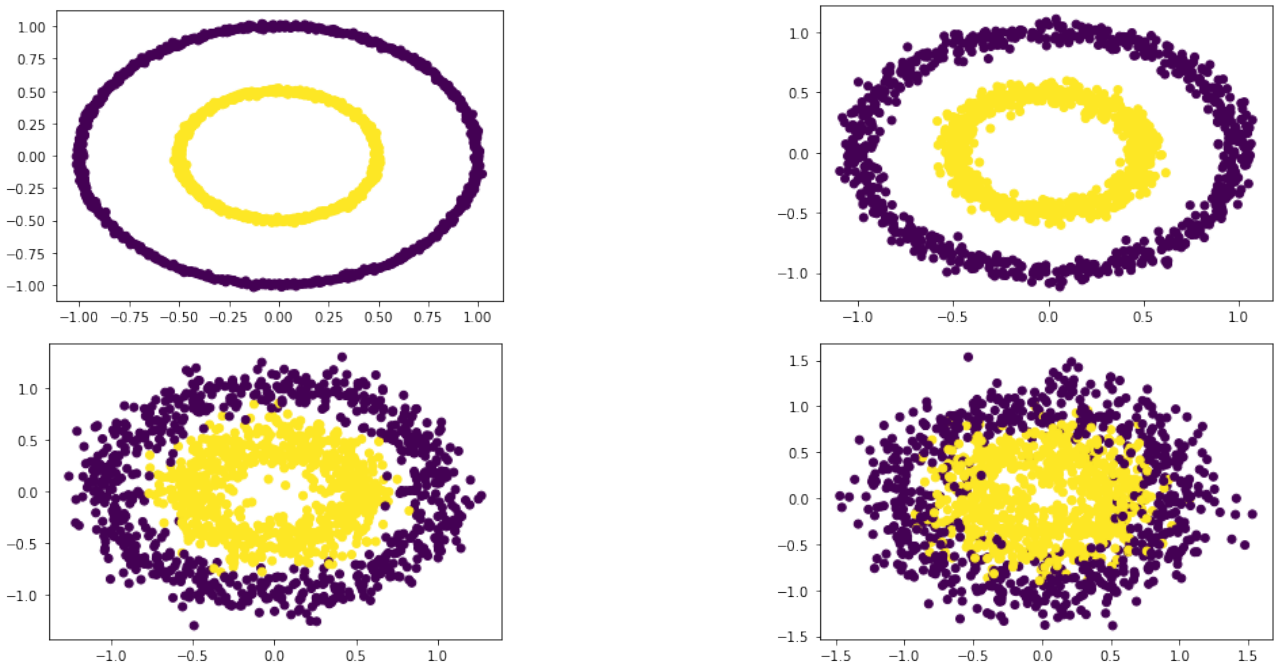
3 Część 2

3.1 Wstęp

W części 2 należało zbadać wpływ parametru *noise* na zachowaniu zbioru danych 2 oraz 3. W obydwu przypadkach badałem go w zakresie od 0.01 do 0.20 co 0.01 (w raporcie tylko 4 wykresy, dla każdego ze zbiorów).

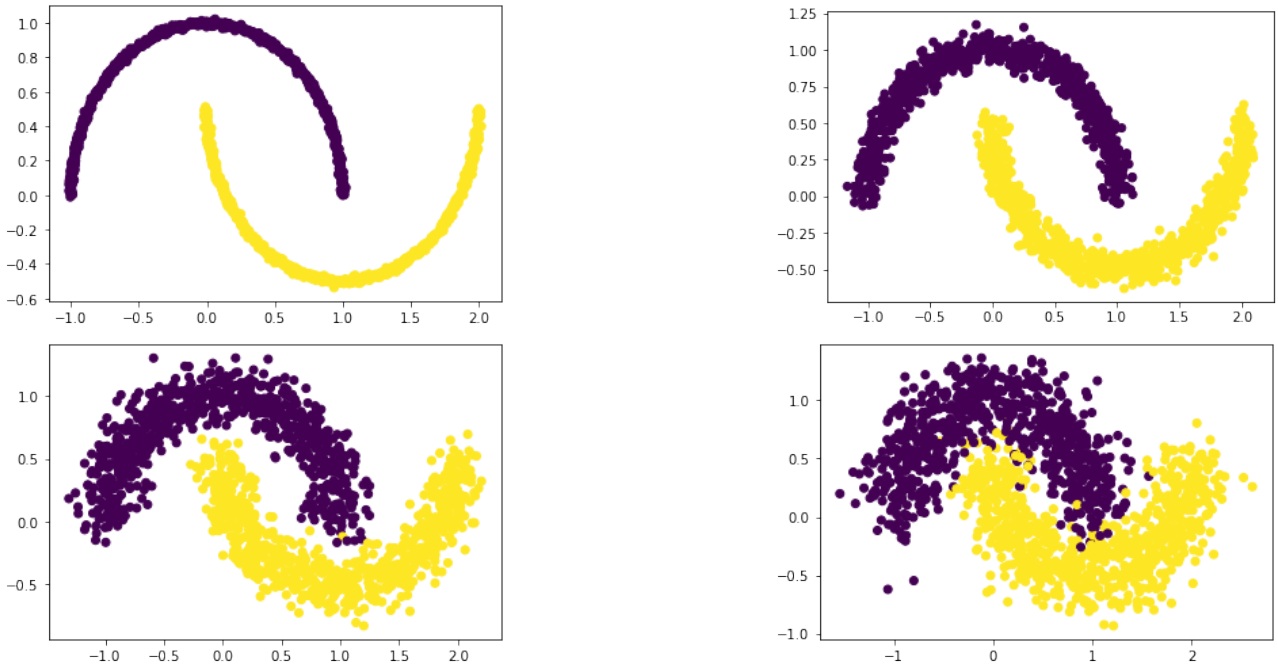
3.2 Zbiory 2 i 3

Dla zbioru danych 2 zmiana parametru daje następujące wyniki:



Rysunek 21: Wizualizacje zbioru danych 2 dla *noise*: 0.01, 0.05, 0.13, 0.19

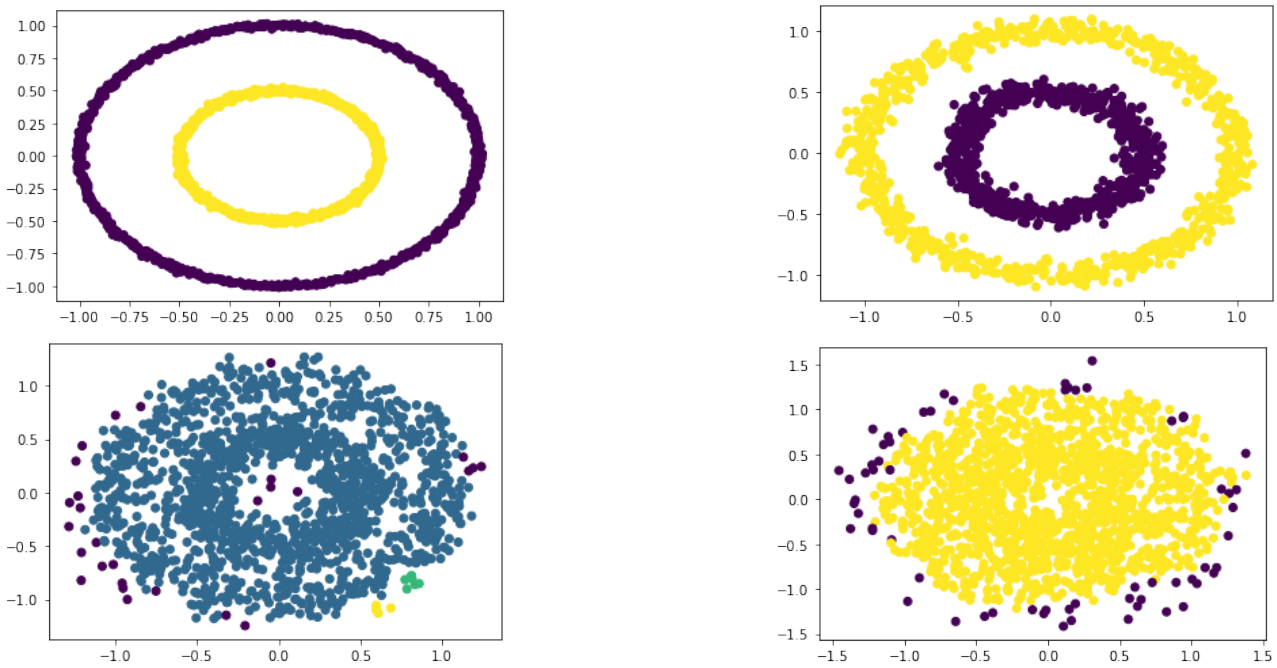
Wraz ze wzrostem parametru *noise* obserwujemy większy rozrzut punktów z obydwu klas. Podobnie sprawa ma się dla zbioru danych 3:



Rysunek 22: Wizualizacje zbioru danych 3 dla *noise*: 0.01, 0.06, 0.12, 0.19

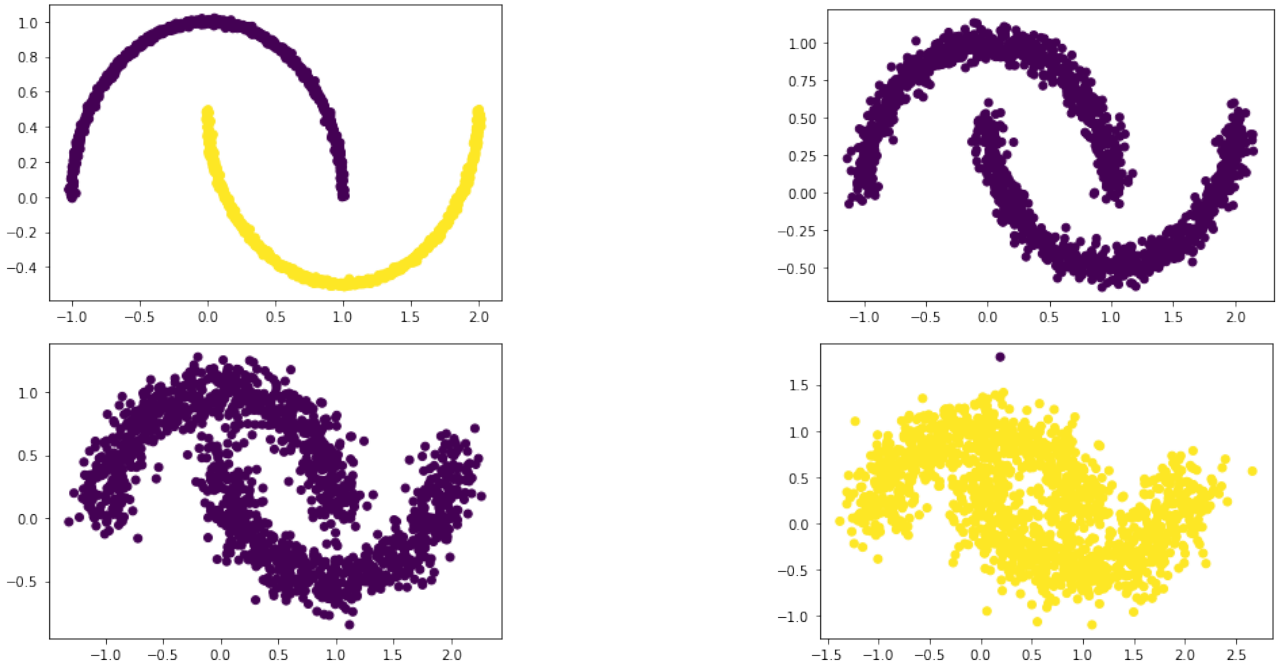
3.3 Klasteryzacja na tych zbiorach

Zbadamy teraz działanie DBSCAN na tych zbiorach pod wpływem zmiany parametru *noise*. Dla danych numer 2 wygląda to następująco:



Rysunek 23: DBSCAN dla zbioru danych 2 z parametrami *eps* = 0.1 i *min_samples* = 5 dla *noise*: 0.01, 0.06, 0.12, 0.19

Dla *noise* równego 0.1 oraz 0.6 klasteryzacja przebiegła pomyślnie. Dla większych wartości większość punktów została zakwalifikowana do jednej klasy.
Dla danych numer 3 prezentuje się to następująco:



Rysunek 24: DBSCAN dla zbioru danych 3 z parametrami *eps* = 0.3 i *min_samples* = 5 dla *noise*: 0.01, 0.06, 0.12, 0.19

Tylko dla *noise* = 0.01 podział został zachowany, w innych przypadkach mamy zakwalifikowanie wszystkich punktów do jednej klasy (poza ostatnim rysunkiem).

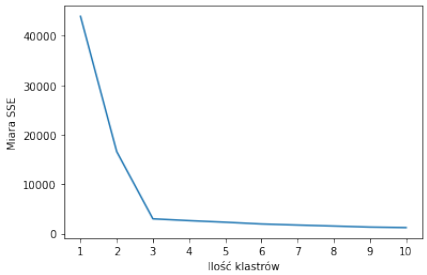
4 Część 3

4.1 Treść

Ostatnim zadaniem było wygenerowanie wykresów pokazującego zależność całkowitej sumy odległości między punktami w klastrach od liczby klastrów. Badanie przeprowadzimy dla modelu KMeans(). Będziemy patrzeć na całkowitą odległość punktów w klasie od centroidów (miara SSE).

4.2 Model 1

W pierwszym modelu wykres ten prezentuje się następująco:

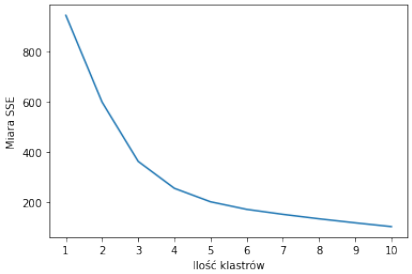


Rysunek 25: Wykres zależności miary SSE od ilości klastrów w zbiorze danych 1.

Na podstawie wykresu, możemy ocenić odpowiednią wartość parametru k (liczby klastrów). Najwięcej zyskujemy przy k = 3, kolejne zmiany są marginalne, stąd wnioskujemy, że jest to najodpowiedniejszy parametr.

4.3 Model 2

W drugim modelu wykres ten wygląda:

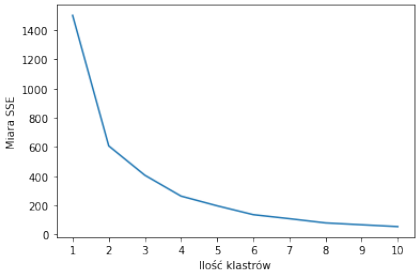


Rysunek 26: Wykres zależności miary SSE od ilości klastrów w zbiorze danych 2.

W tym przypadku biorąc pod uwagę zadanie 1, metoda KMeans nie poradziła sobie najlepiej z podziałem na klastry. Oceniając ilość klastrów jedynie na podstawie tego wykresu można stwierdzić, że odpowiednim parametrem byłoby k = 4.

4.4 Model 3

Dla trzeciego modelu wykres przedstawia się tak oto:

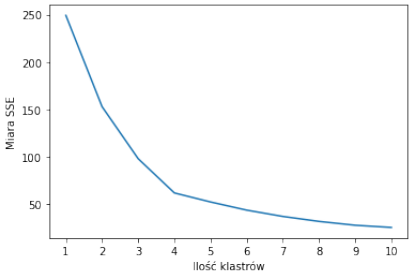


Rysunek 27: Wykres zależności miary SSE od ilości klastrów w zbiorze danych 3.

Ponownie w tym przypadku w zadaniu 1 KMeans nie poradziło sobie najlepiej. Uwzględniając tylko wykres, optymalnym parametrem k byłoby k = 2.

4.5 Model 4

W ostatnim zbiorze danych wygląda to nastepująco:



Rysunek 28: Wykres zależności miary SSE od ilości klastrów w zbiorze danych 4.

Dla danych losowych najbardziej odpowiedni wydaje się parametr k = 4.

5 Wnioski

Wyniki z zadania 1 zależą bardzo od użytej metody. Dla zbioru danych typu 1 dobrze wypadło KMeans, natomiast w przypadku zbiorów danych 2 i 3 poprawne rezultaty dawało DBSCAN. Należy pamiętać o odpowiednim doborze liczby klastrów, czy w metodzie DBSCAN parametrów *eps* oraz *min_samples*. W zadaniu 2 parametr *noise* powodował większy rozrzut w danych. Metoda DBSCAN, która radziła sobie w poprzedniej części, w tej dla dużego szumu okazała się nieskuteczna. Natomiast postać wykresów z zadania 3 pozwala oszacować parametr k.