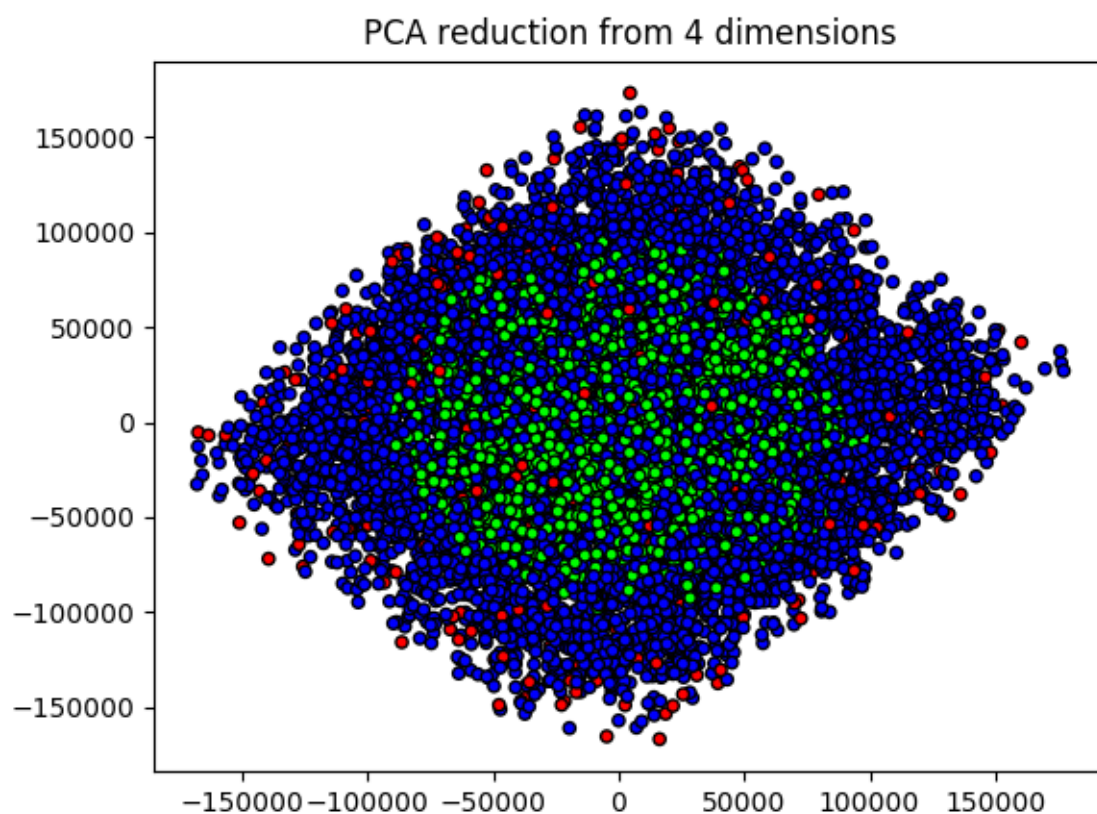
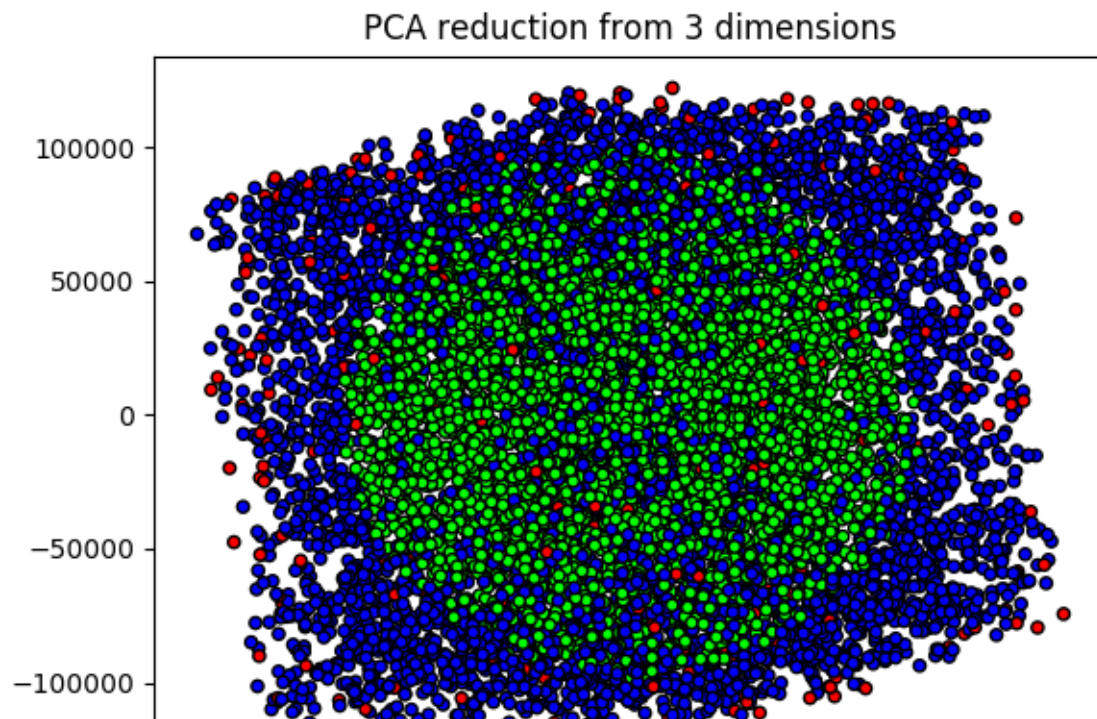


Metody Rozpoznawania Obrazów I Podstawy Uczenia Maszynowego

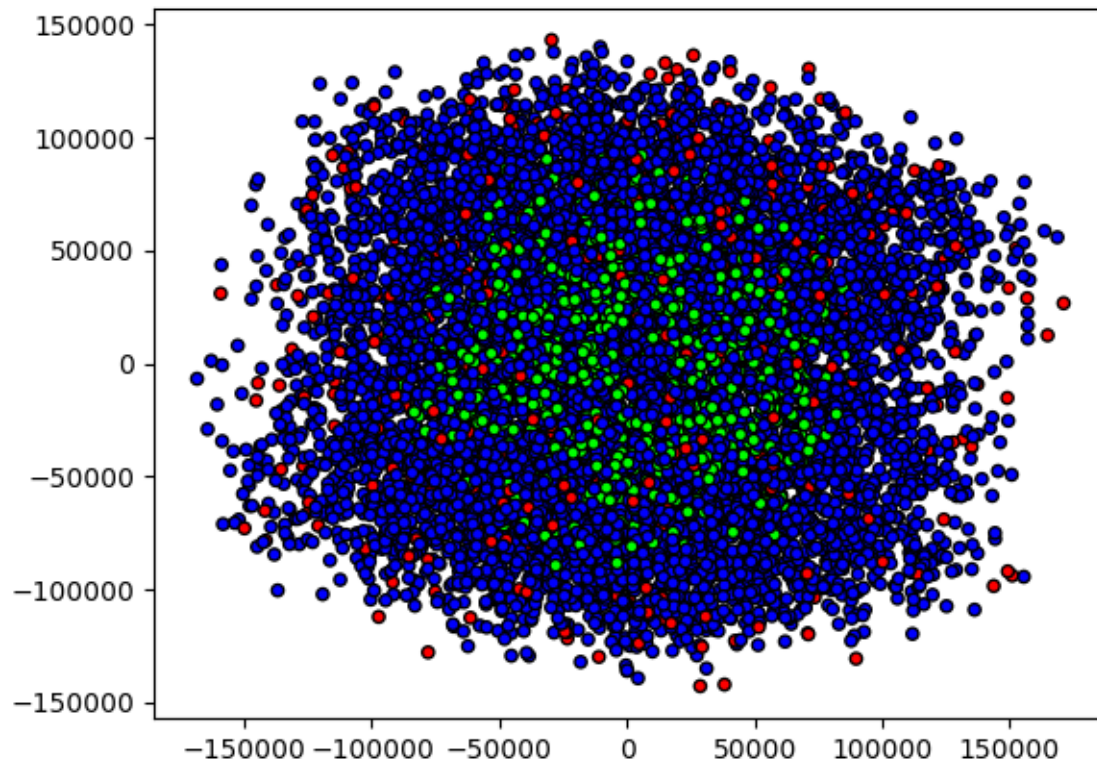
Kernel trick

Autor: Ryszard Sikora

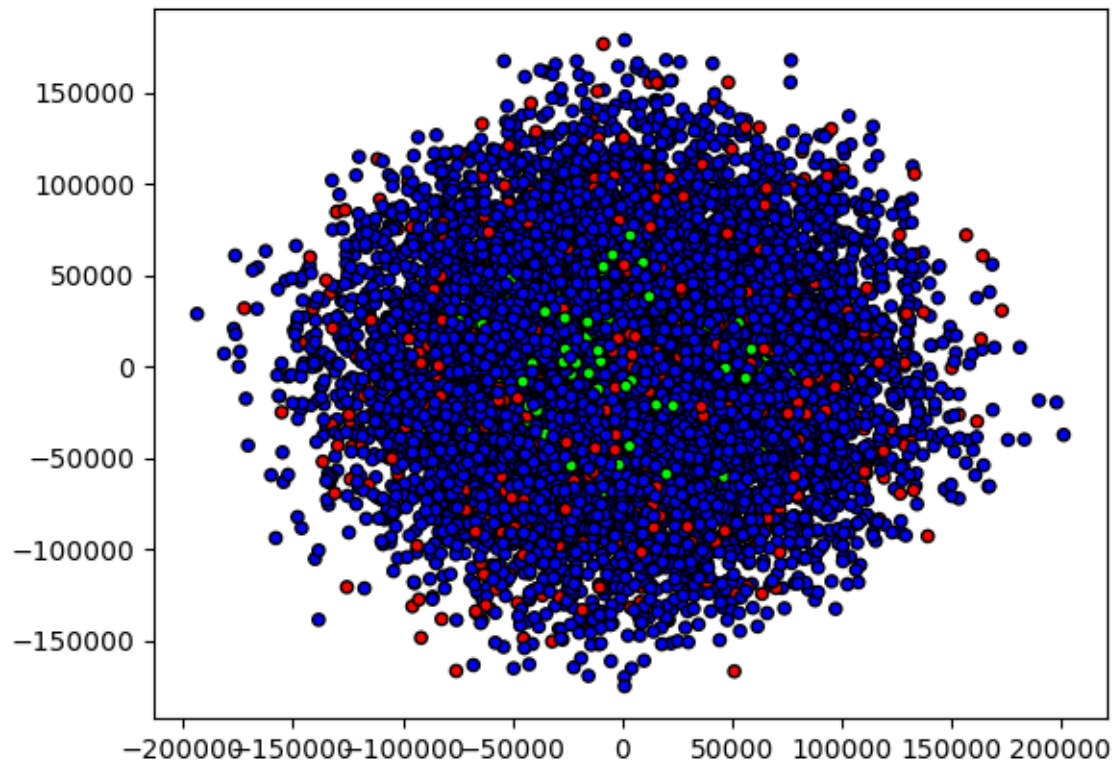
Zadanie A

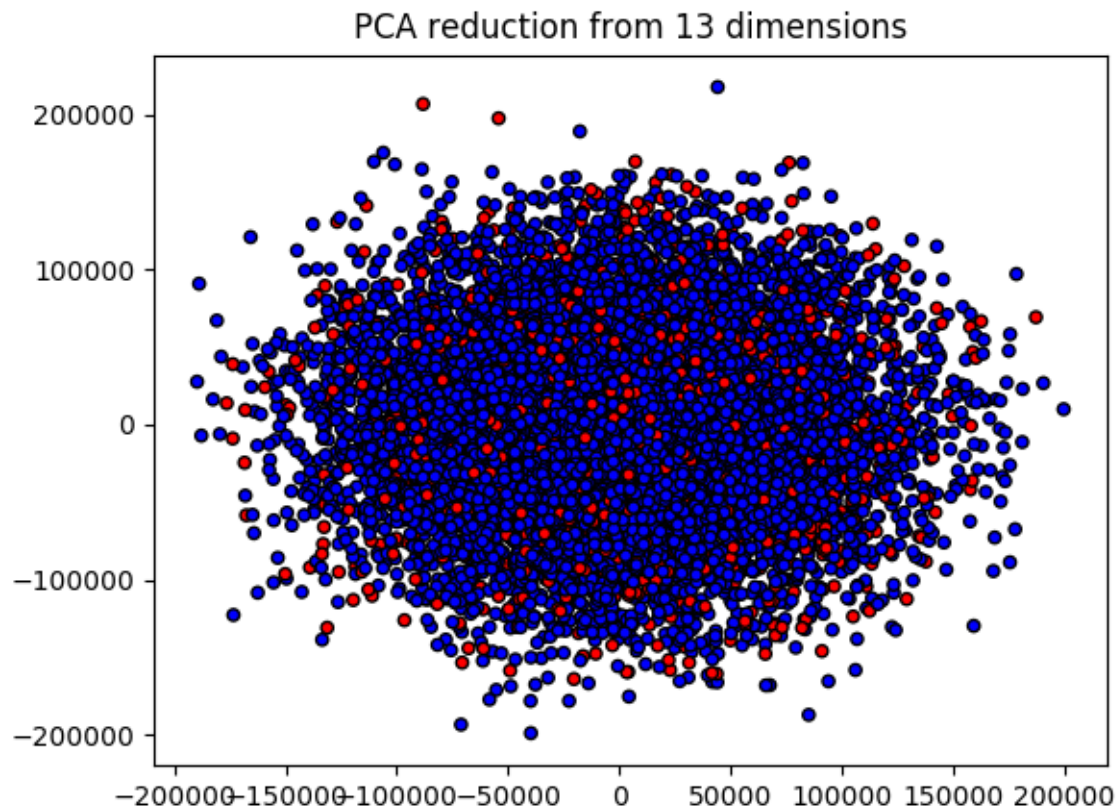


PCA reduction from 5 dimensions



PCA reduction from 7 dimensions





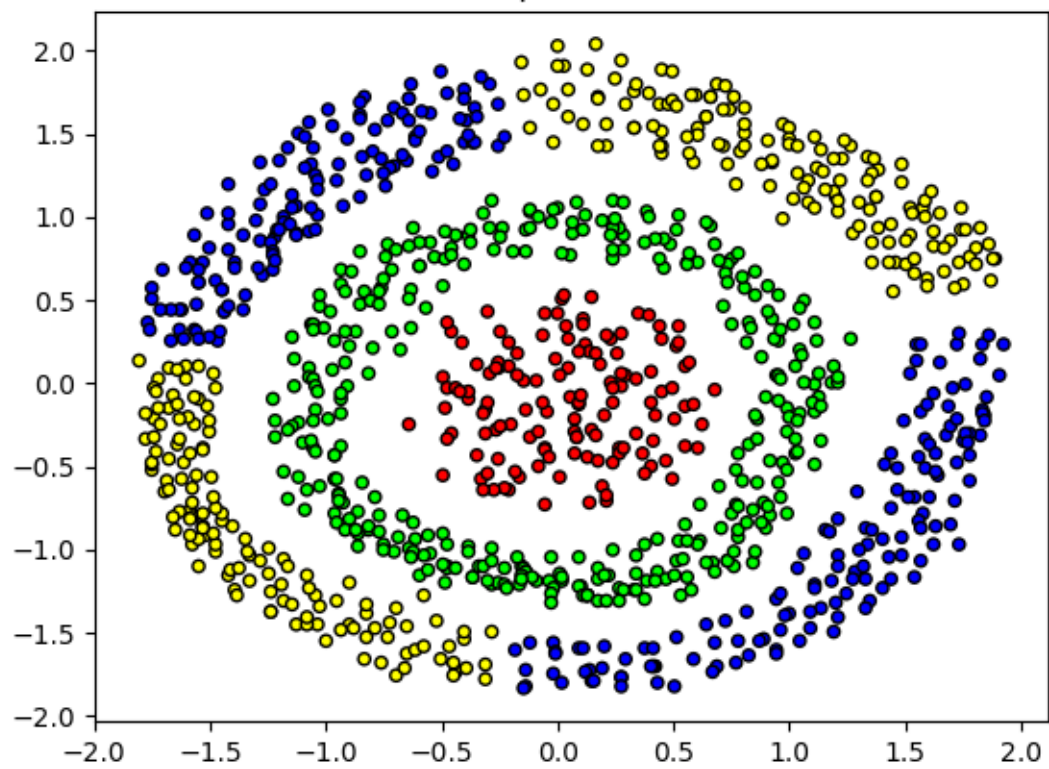
Punktów wewnątrz kuli jest coraz mniej – wynika to z coraz mniejszego stosunku objętości hiperkuli wpisanej w hipersześcian. Czerwonych jest dużo – trochę sztucznie zawyżyłem ich ilość, bo bez tego się nie losowały.

Zadanie B

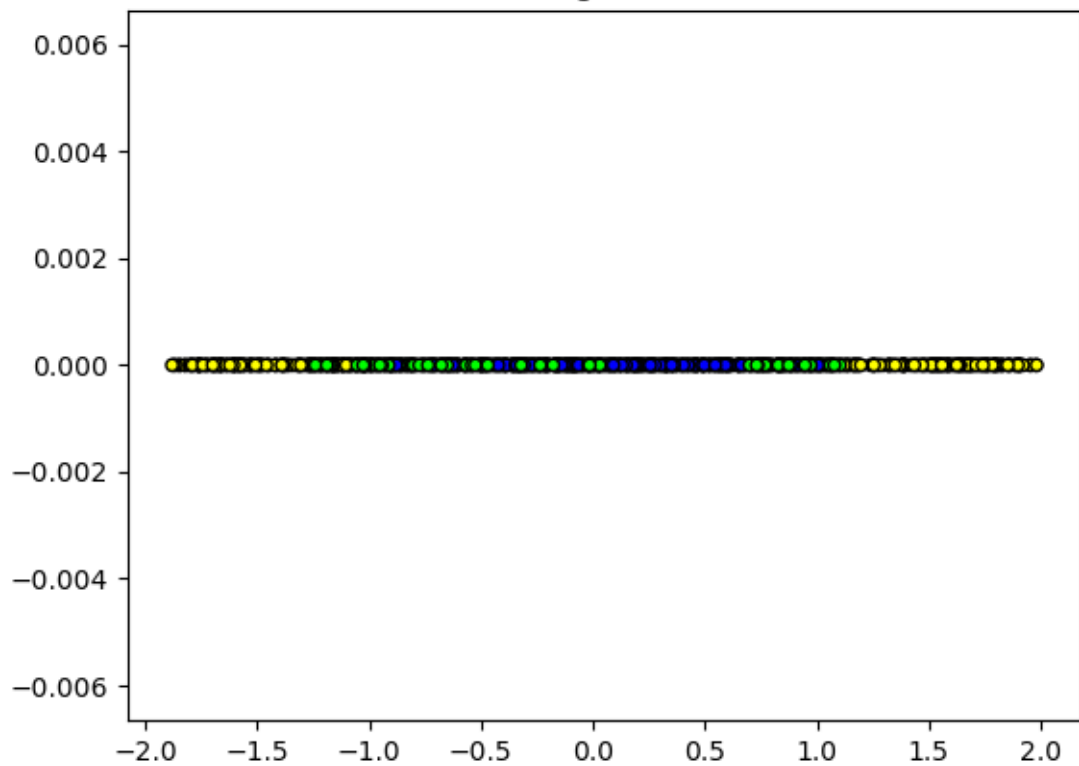
Dane wygenerowałem metodą painta.

Pierwsza próba z wycentrowaniem danych.

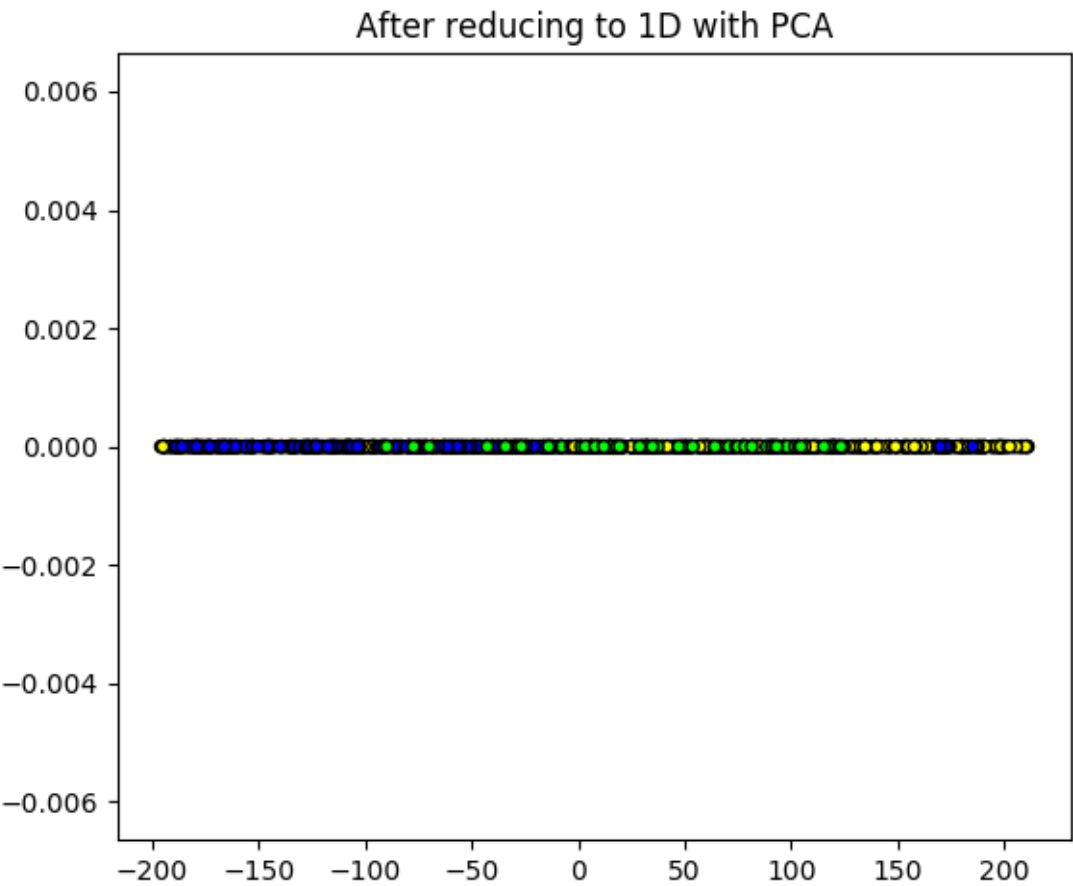
Input data



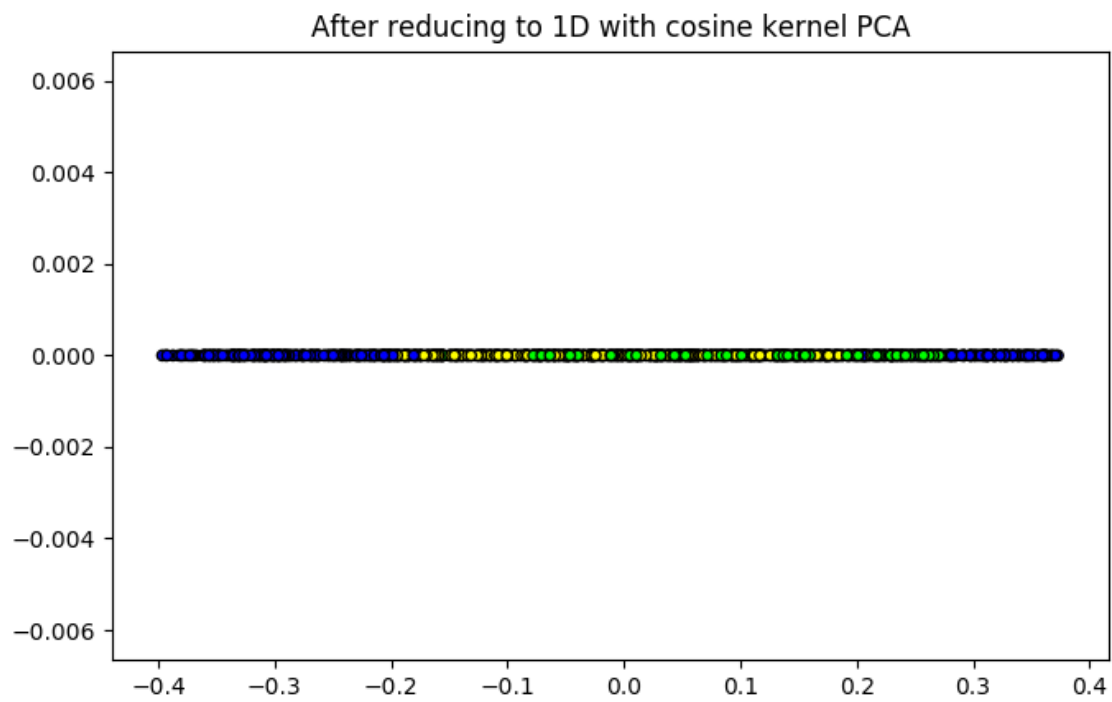
After reducing to 1D with PCA



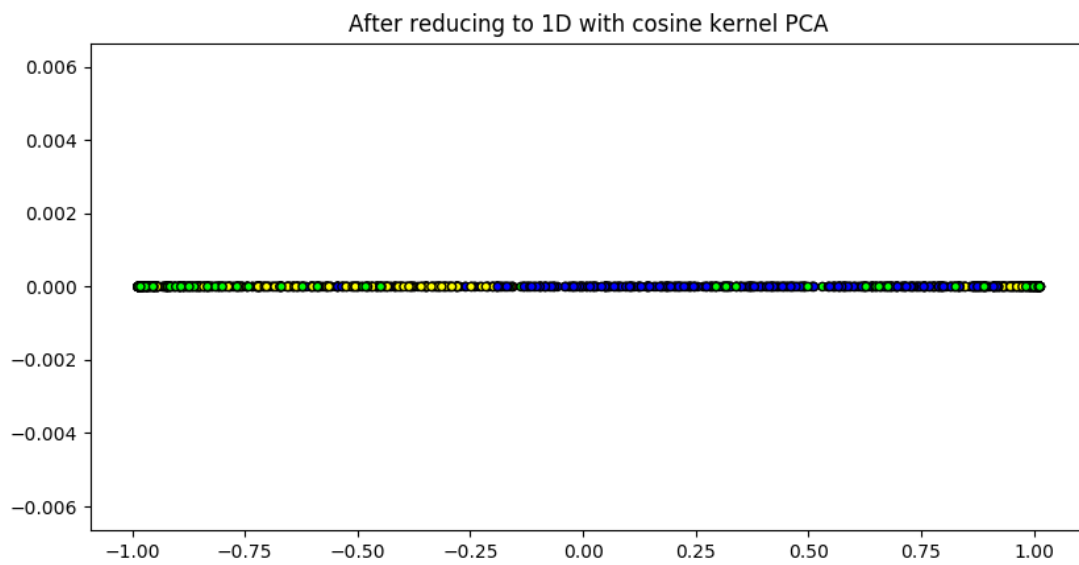
Druga próba, bez centrowania danych.



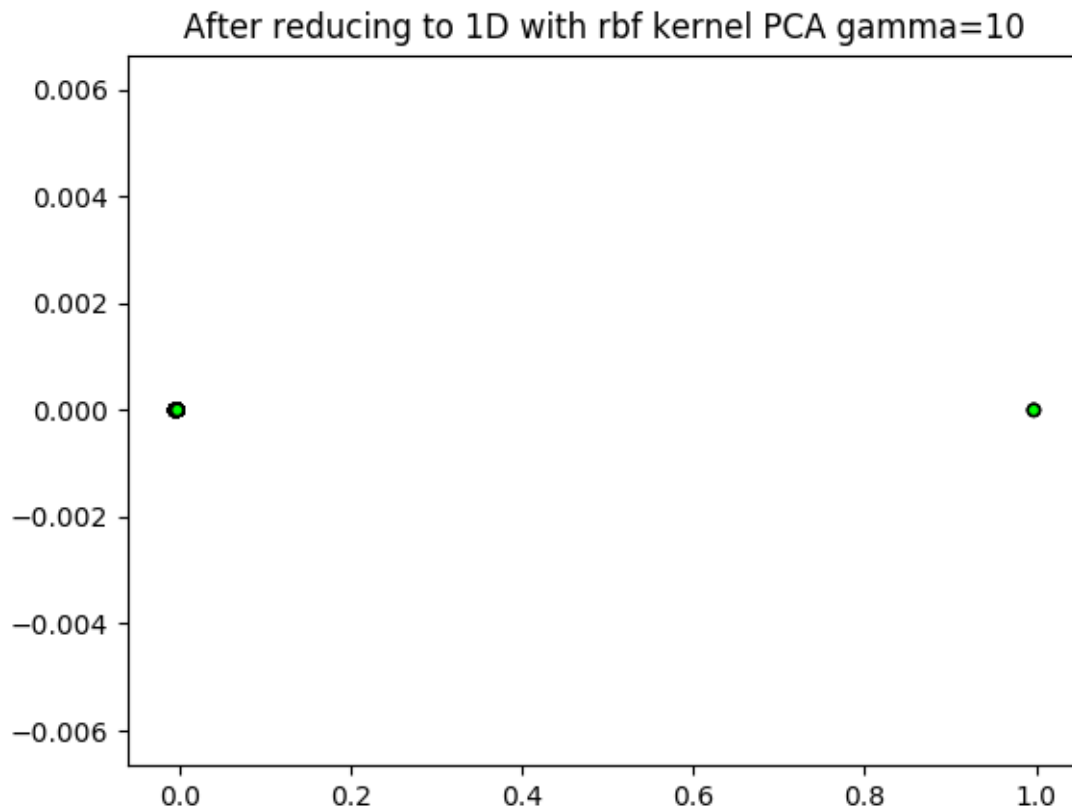
Cosine KernelPCA bez centrowania danych



Cosine kernel PCA z centrowaniem danych

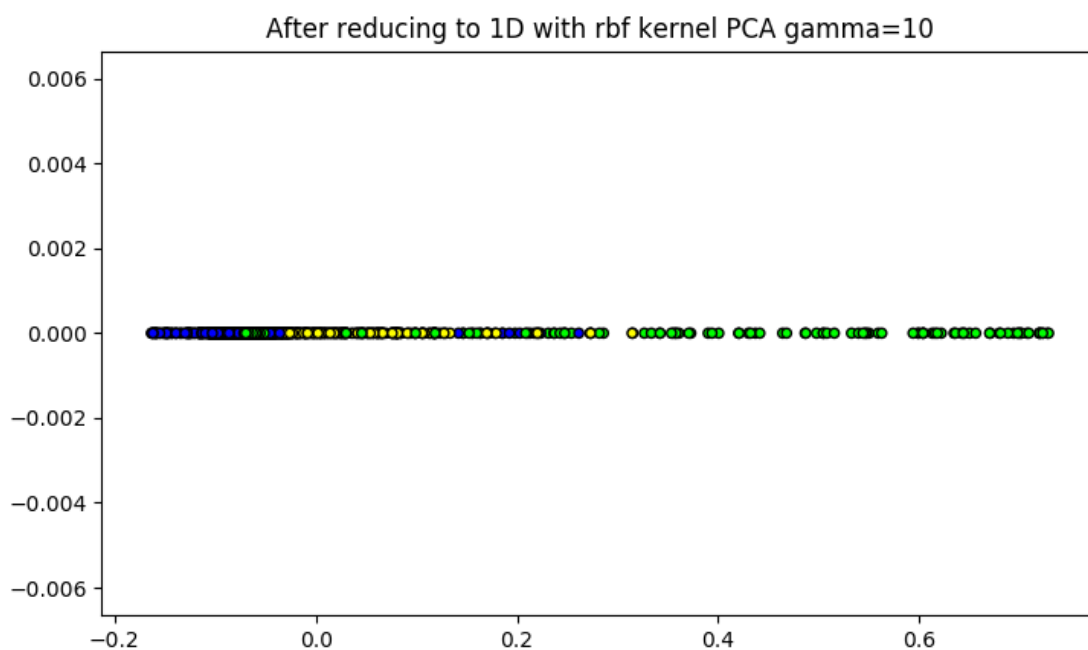


Rbf kernel PCA bez centrowania danych



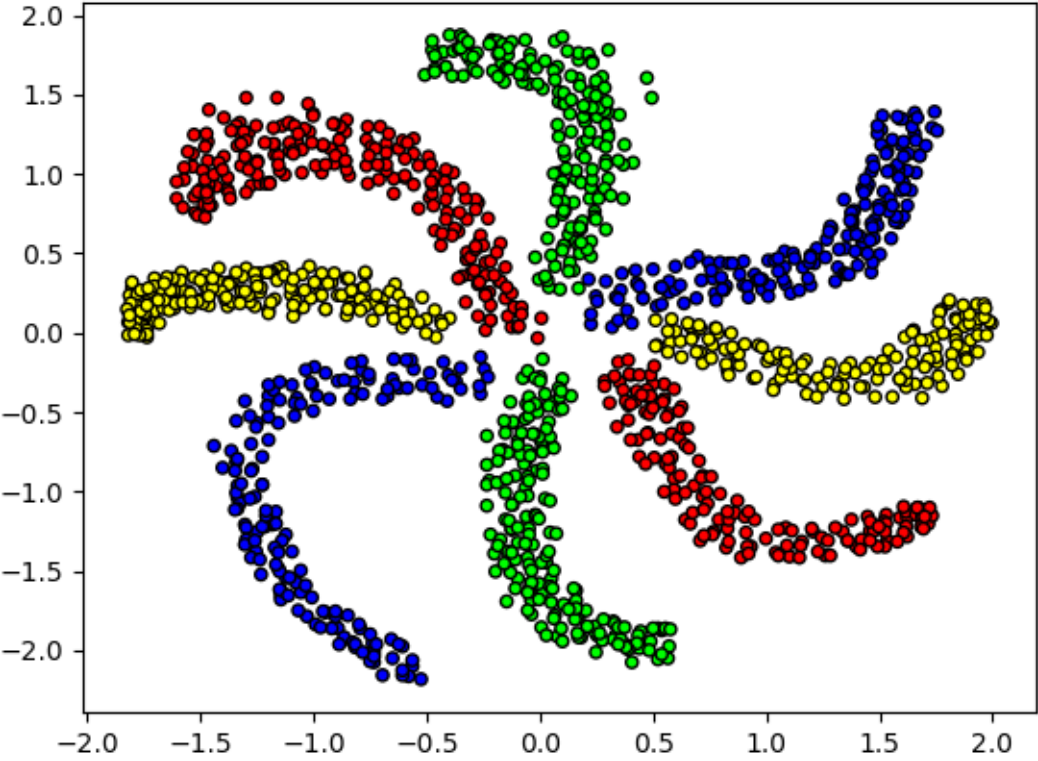
To chyba jakiś błąd wynikający z braku centrowania danych.

Rbf kernel PCA z centrowaniem danych

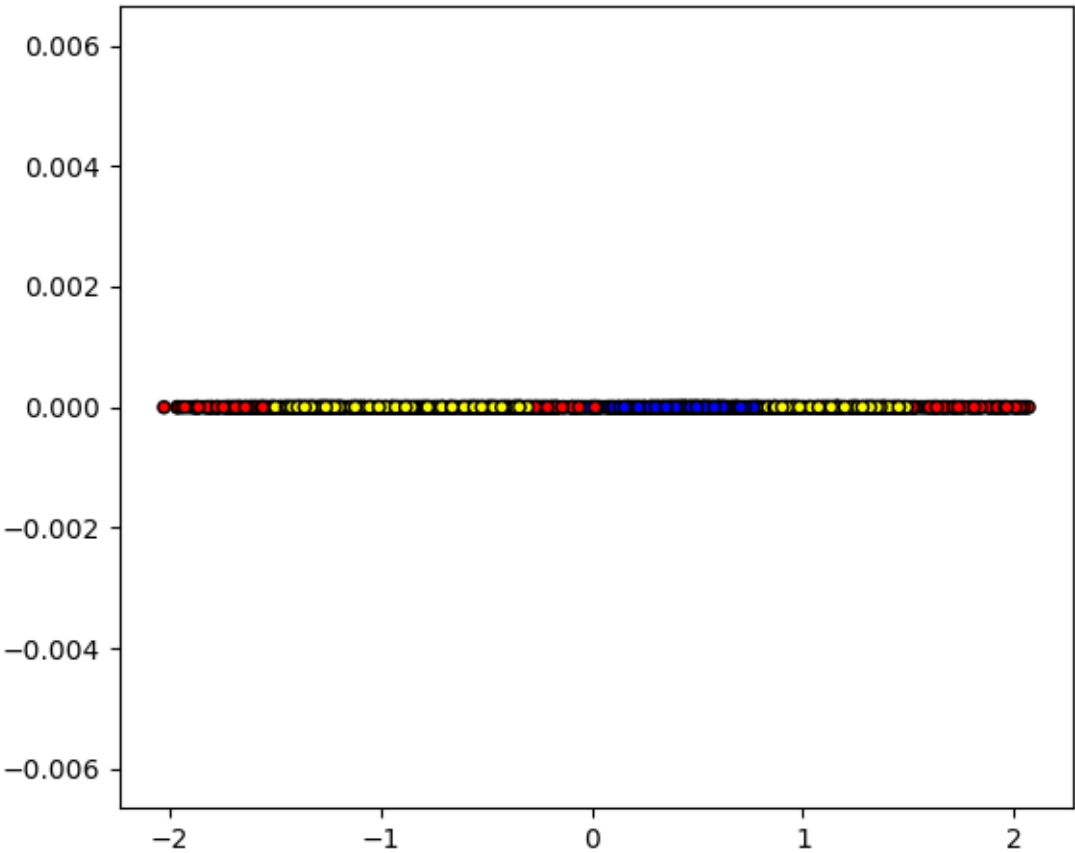


Widać, że zastosowanie kernel PCA poprawiło rezultaty redukcji. Natomiast wciąż nie jest to idealna redukcja – gdzieś nam zjadło czerwone punkty.

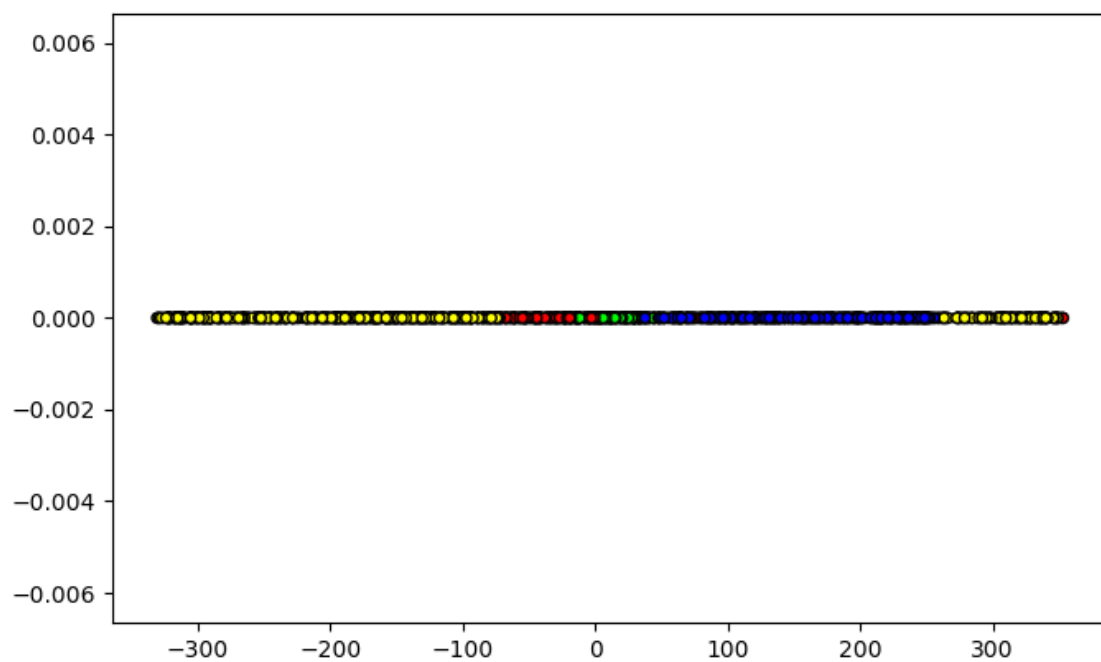
Pora na drugi zestaw danych.
Scentralizowane dane wejściowe



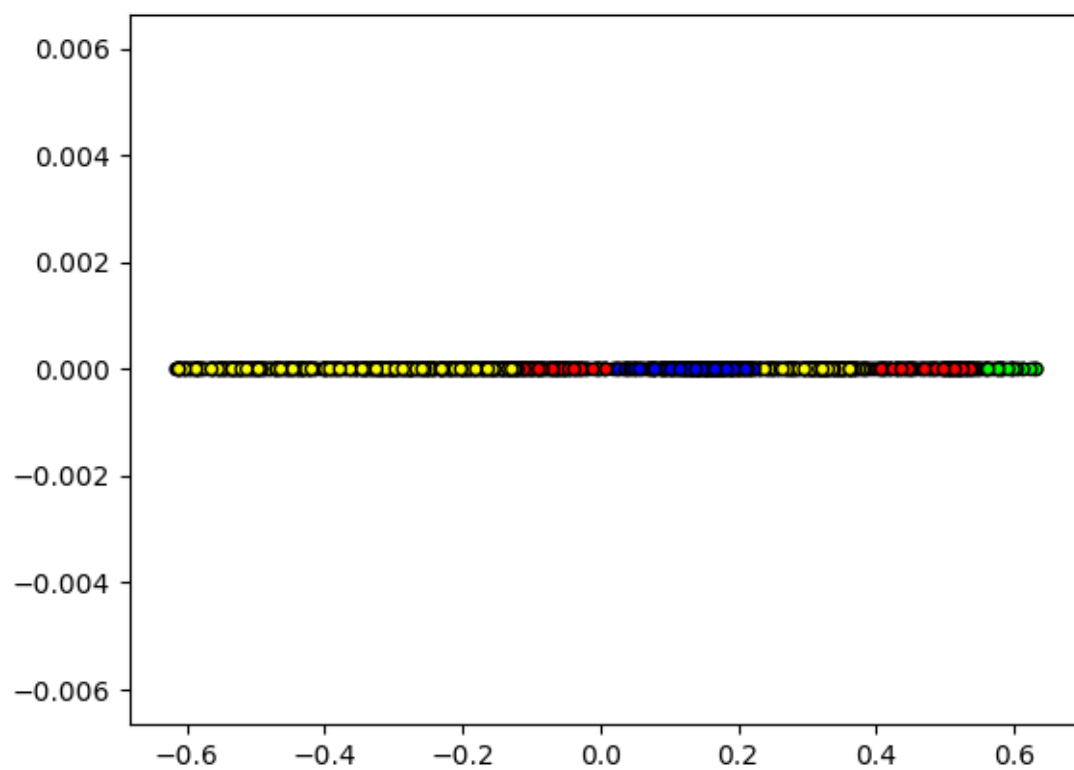
PCA dla centrowanych danych



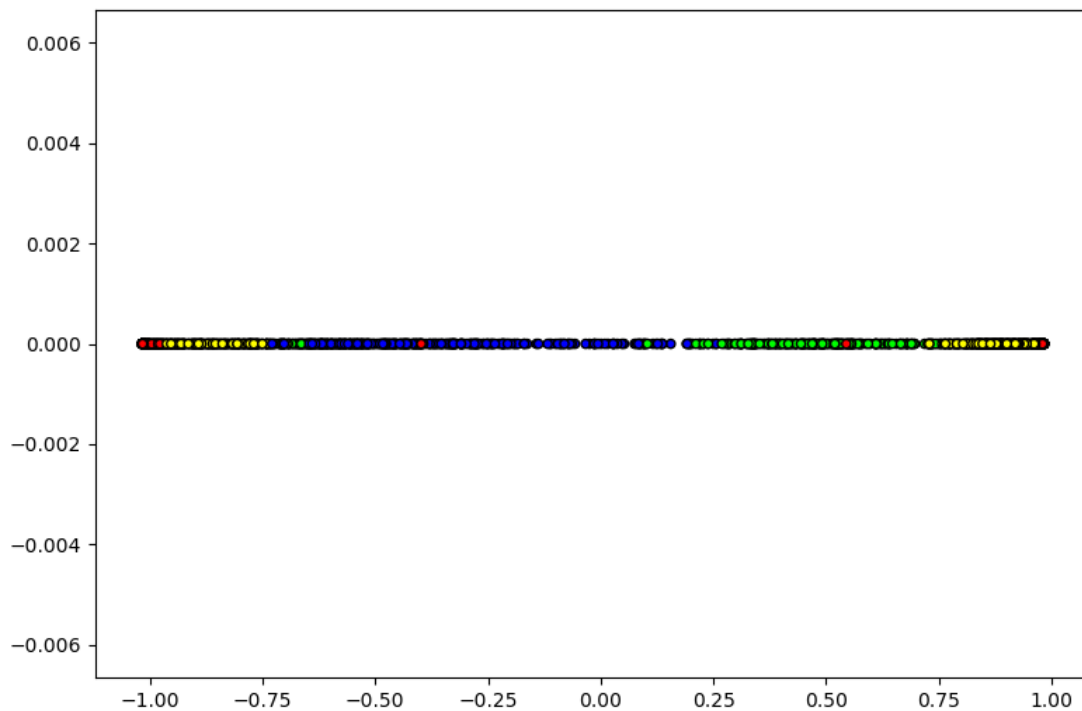
PCA dla niescentrowanych danych



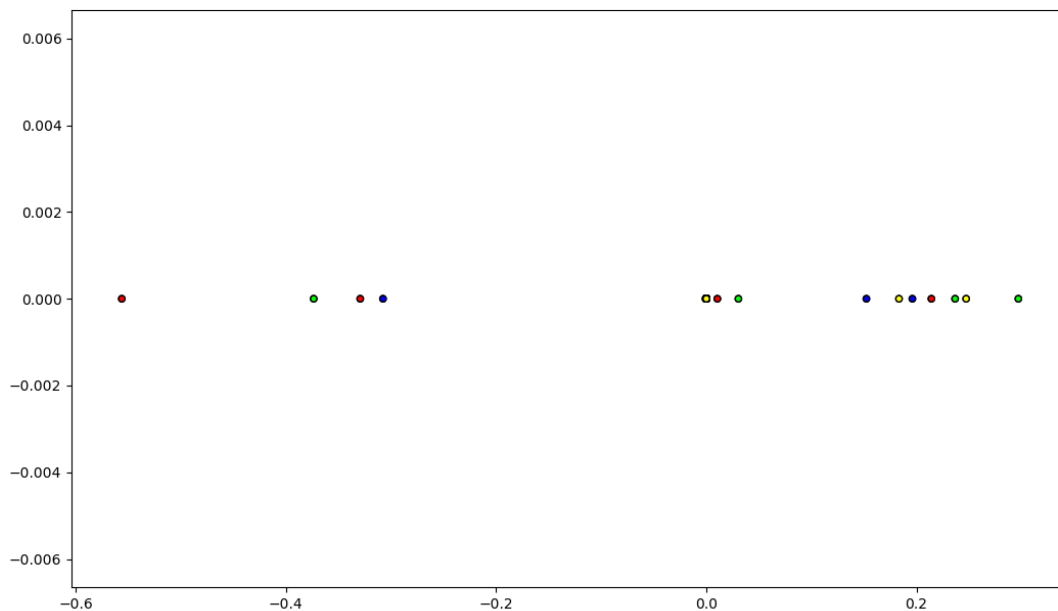
cosine kernel PCA dla niescentrowanych danych



cosine kernel PCA dla scentrowanych danych

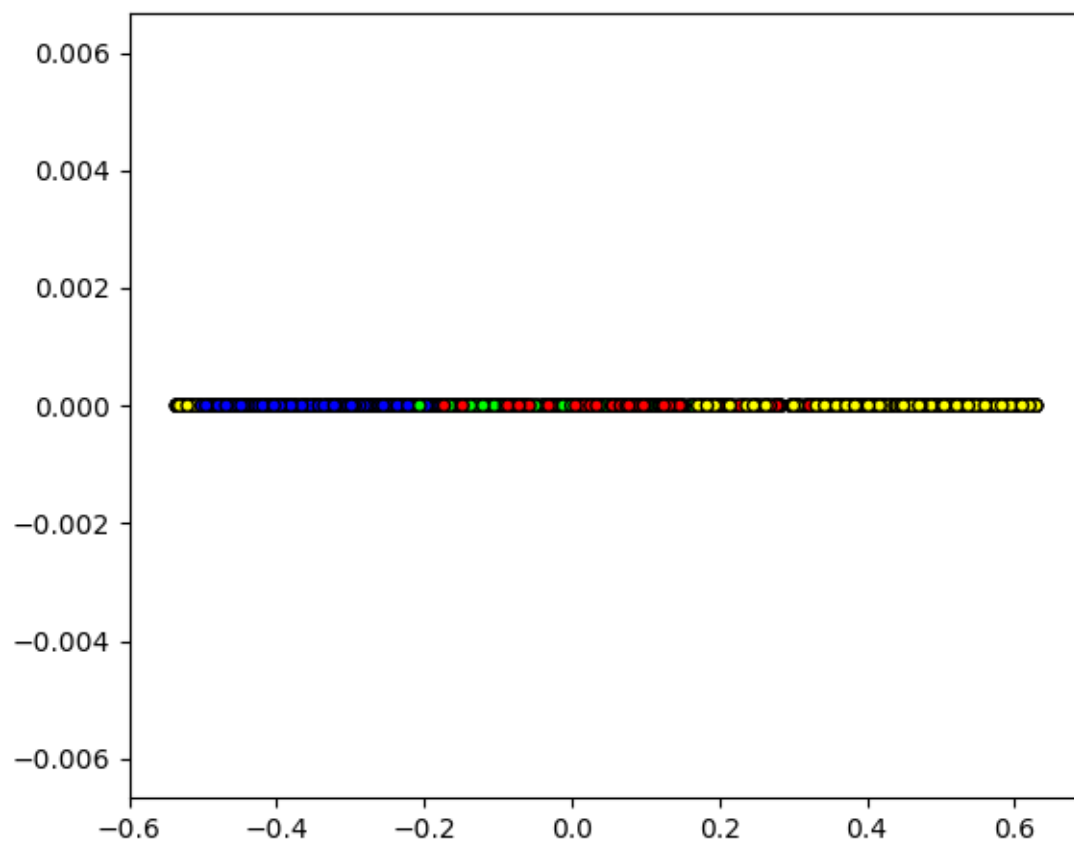


rbf kernel PCA dla niescentrowanych danych, gamma = 15

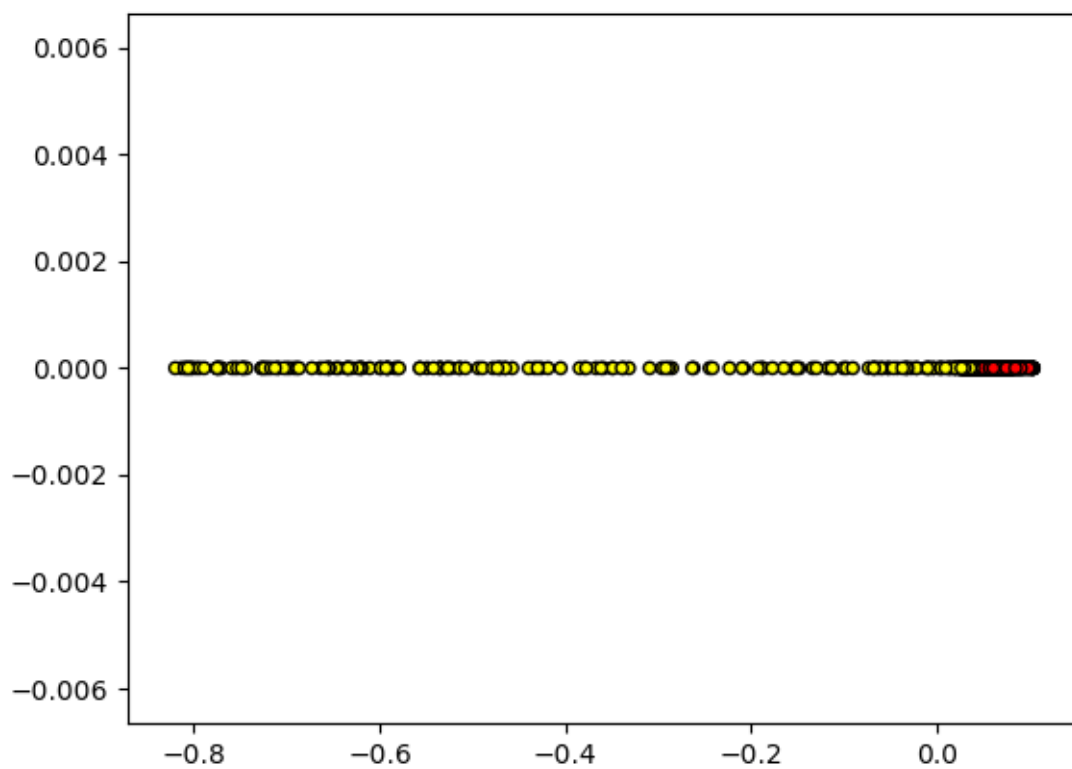


Trudno stwierdzić czy metoda znalazła idealny podział, że większość punktów wylądowała w tym samym miejscu, bo klasy niby wszystkie są, ale z drugiej strony dla tego samego przypadku poprzedni przykład wyświetlił na pewno błędny podział.

Rbf kernel PCA dla scentrowanych danych, $\gamma = 2$

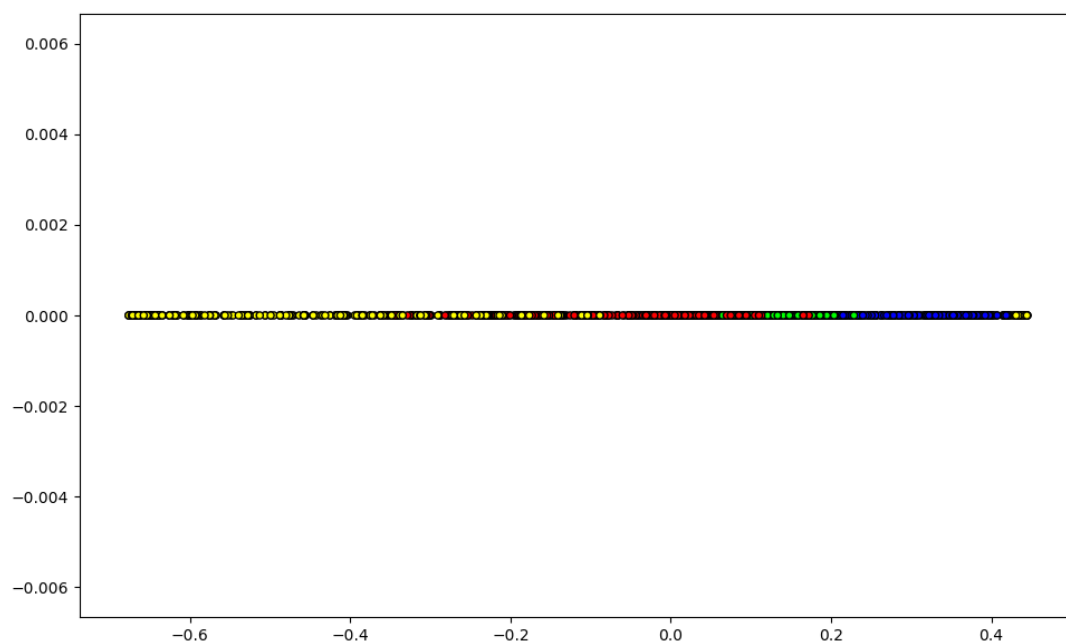


Rbf kernel PCA dla scentrowanych danych, $\gamma = 15$



Parametr gamma zdecydowanie za duży. Poprzedni przykład lepiej się spisał.

Rbf kernel PCA dla scentrowanych danych, $\gamma = 4$



Subiektywnie, najlepszy wynik dla $\gamma = 4$.

Jak widać użycie kernel trick może zdecydowanie poprawić działanie PCA, aczkolwiek są przypadki, w których nawet to nie wystarczy i trzeba skorzystać z innego algorytmu.