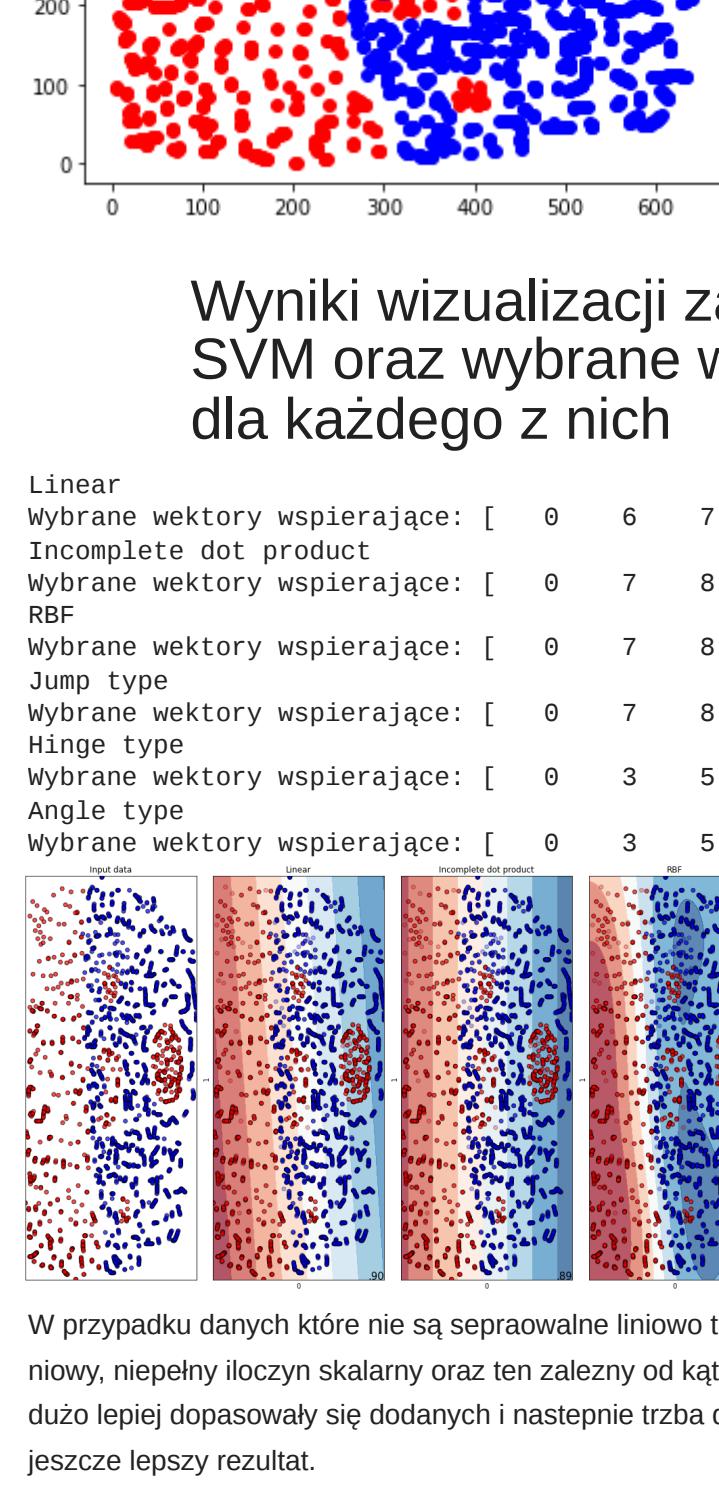


Wizualizacja zbioru danych



Wyniki wizualizacji zadanych wariantów SVM oraz wybrane wektory wspierające dla każdego z nich

Linear

Wybrane wektory wspierające: [0 6 7 ... 2893 2895 2900]

Incomplete dot product

Wybrane wektory wspierające: [0 7 8 ... 2893 2895 2900]

RBF

Wybrane wektory wspierające: [0 7 8 ... 2886 2889 2895]

Jump type

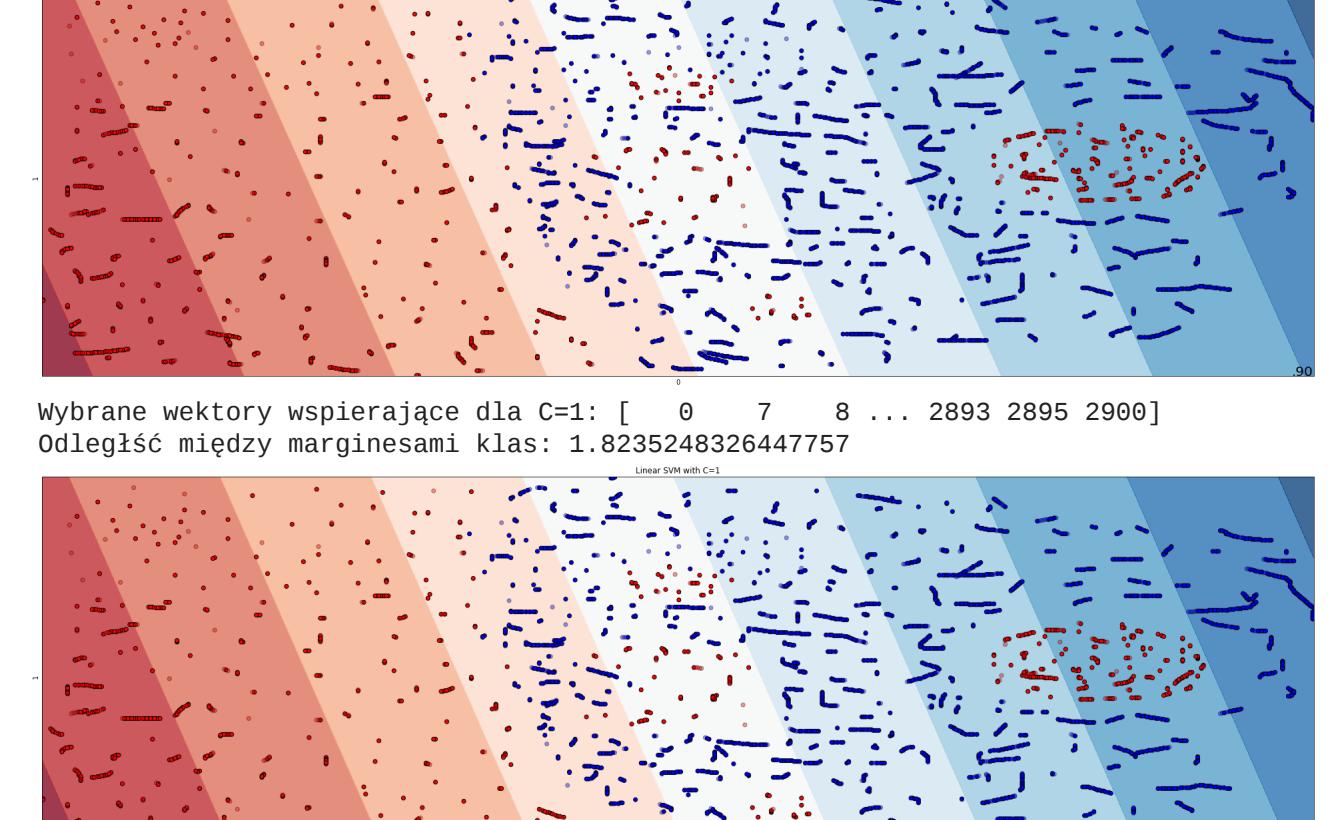
Wybrane wektory wspierające: [0 7 8 ... 2886 2889 2895]

Hinge type

Wybrane wektory wspierające: [0 3 5 ... 2886 2889 2899]

Angle type

Wybrane wektory wspierające: [0 3 5 ... 2893 2895 2900]



W przypadku danych który nie są seprawalne liniowo tak jak te w zadaniu kernele liniowe tzn. niowy, niepełny iloczyn skalarny oraz ten zależny od kątów nie dają dobrych rezultatów. Pozostałe 3 dużo lepiej dopasowały się dodanych i następnie trzeba dla nich dobrze parametry, żeby dawały jeszcze lepszy rezultat.

Sprawdzanie różnych wartości kary dla liniowego SVM

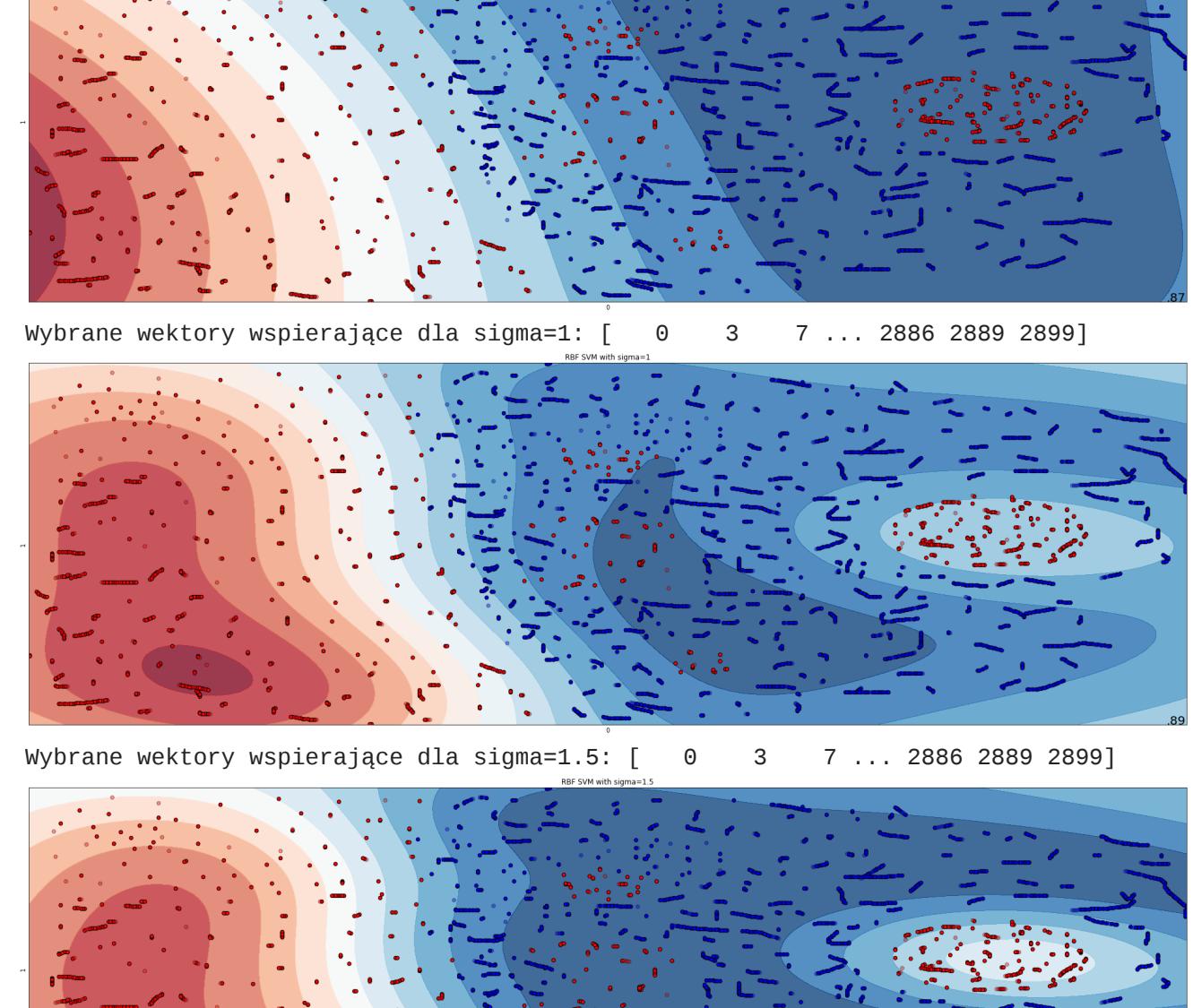
Wybrane wektory wspierające dla C=0.0001: [0 3 5 ... 2895 2897 2900]

Odległość między marginesami klas: 40.44009359361703



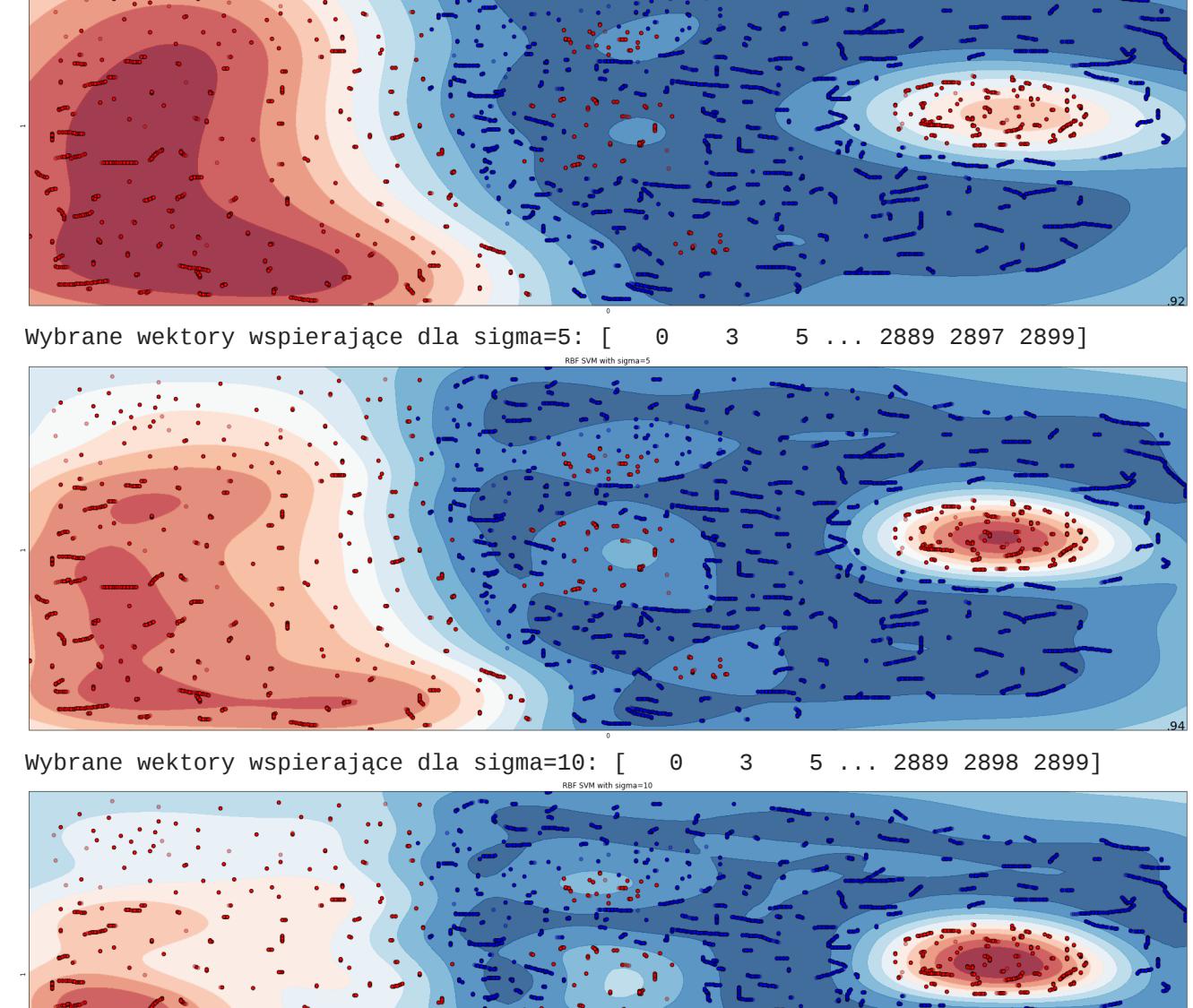
Wybrane wektory wspierające dla C=0.001: [0 3 5 ... 2895 2897 2900]

Odległość między marginesami klas: 4.044217909750173



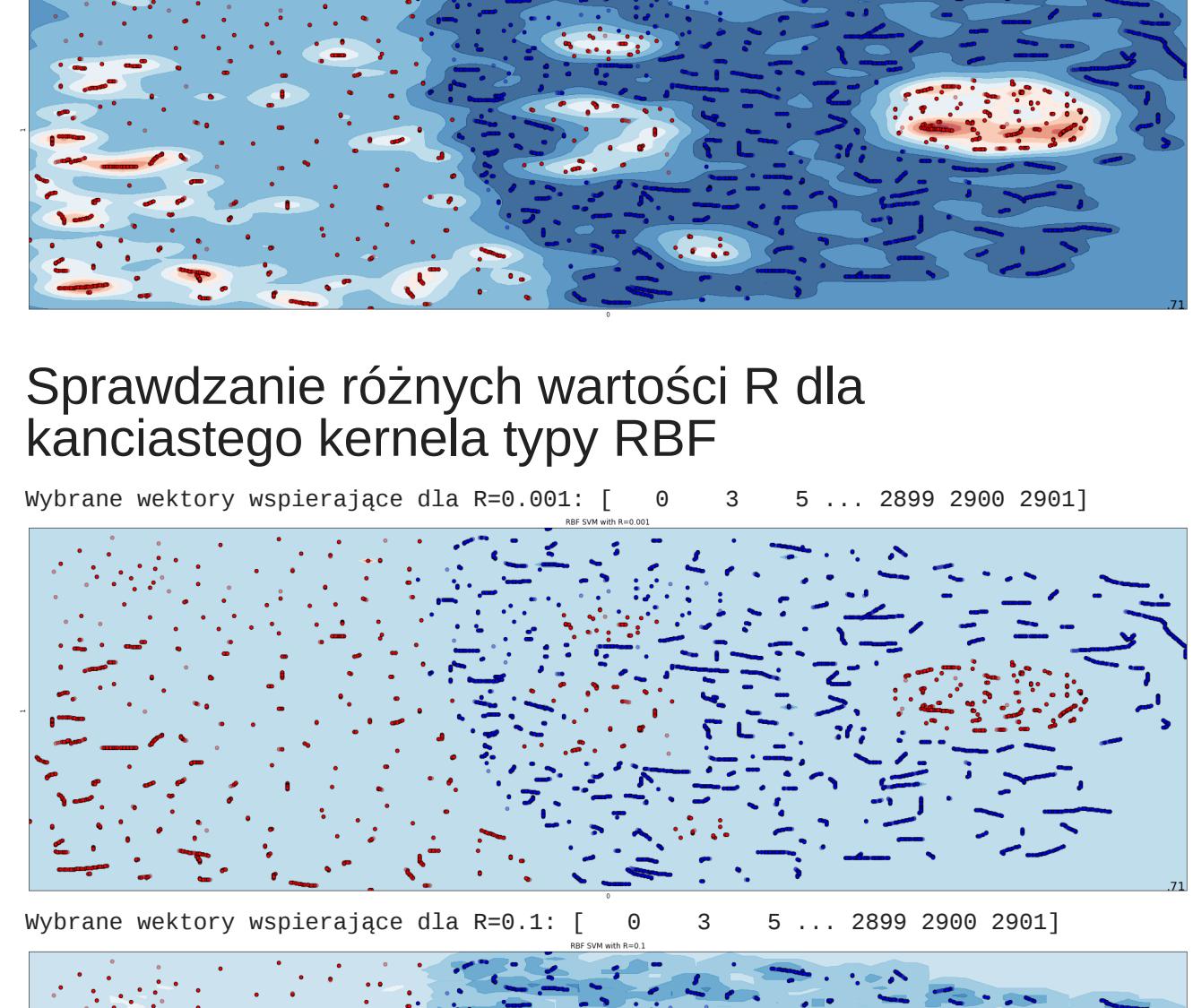
Wybrane wektory wspierające dla C=0.1: [0 7 8 ... 2893 2895 2900]

Odległość między marginesami klas: 1.9533593540029446



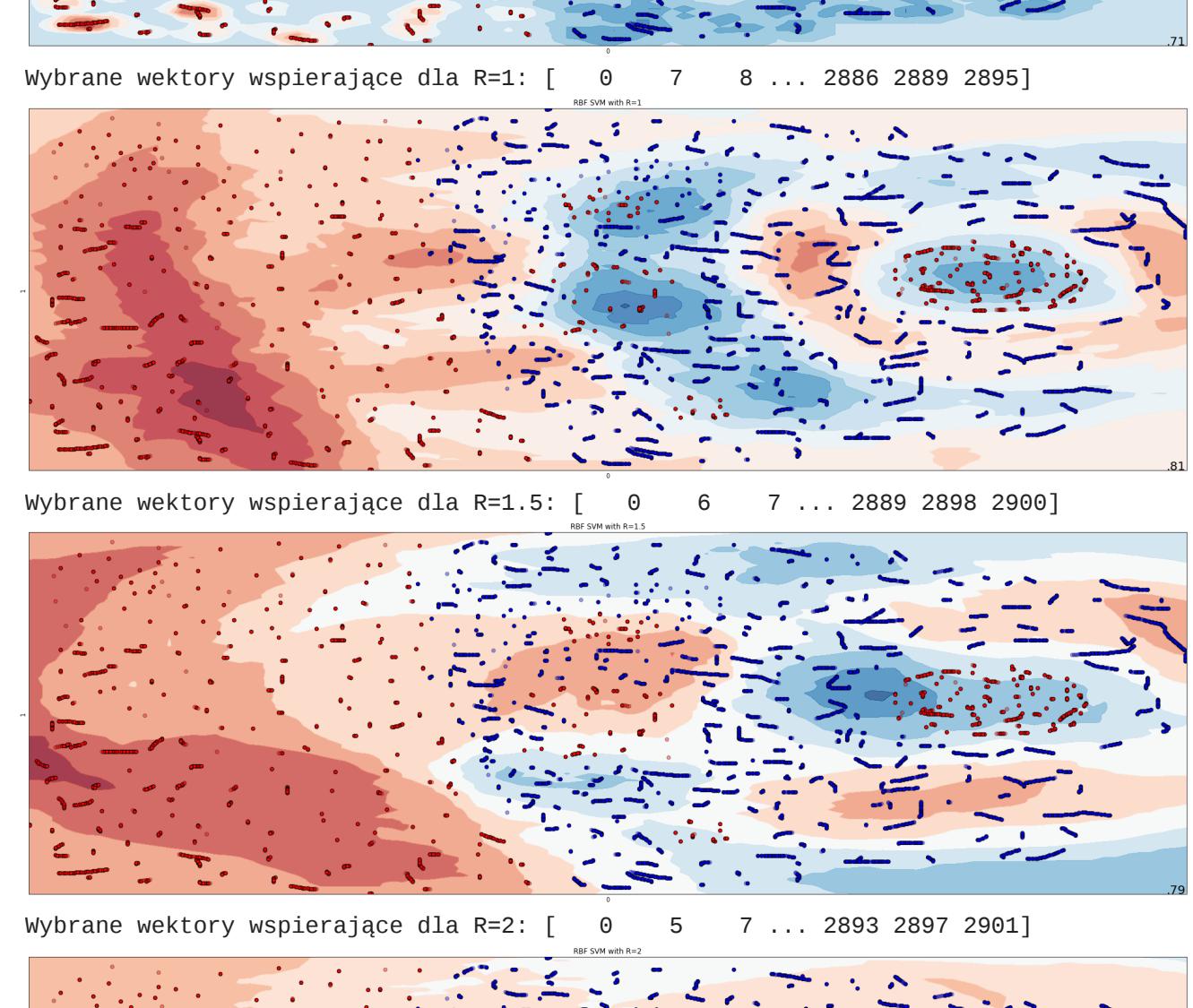
Wybrane wektory wspierające dla C=0.7: [0 7 8 ... 2893 2895 2900]

Odległość między marginesami klas: 1.8266956685128057



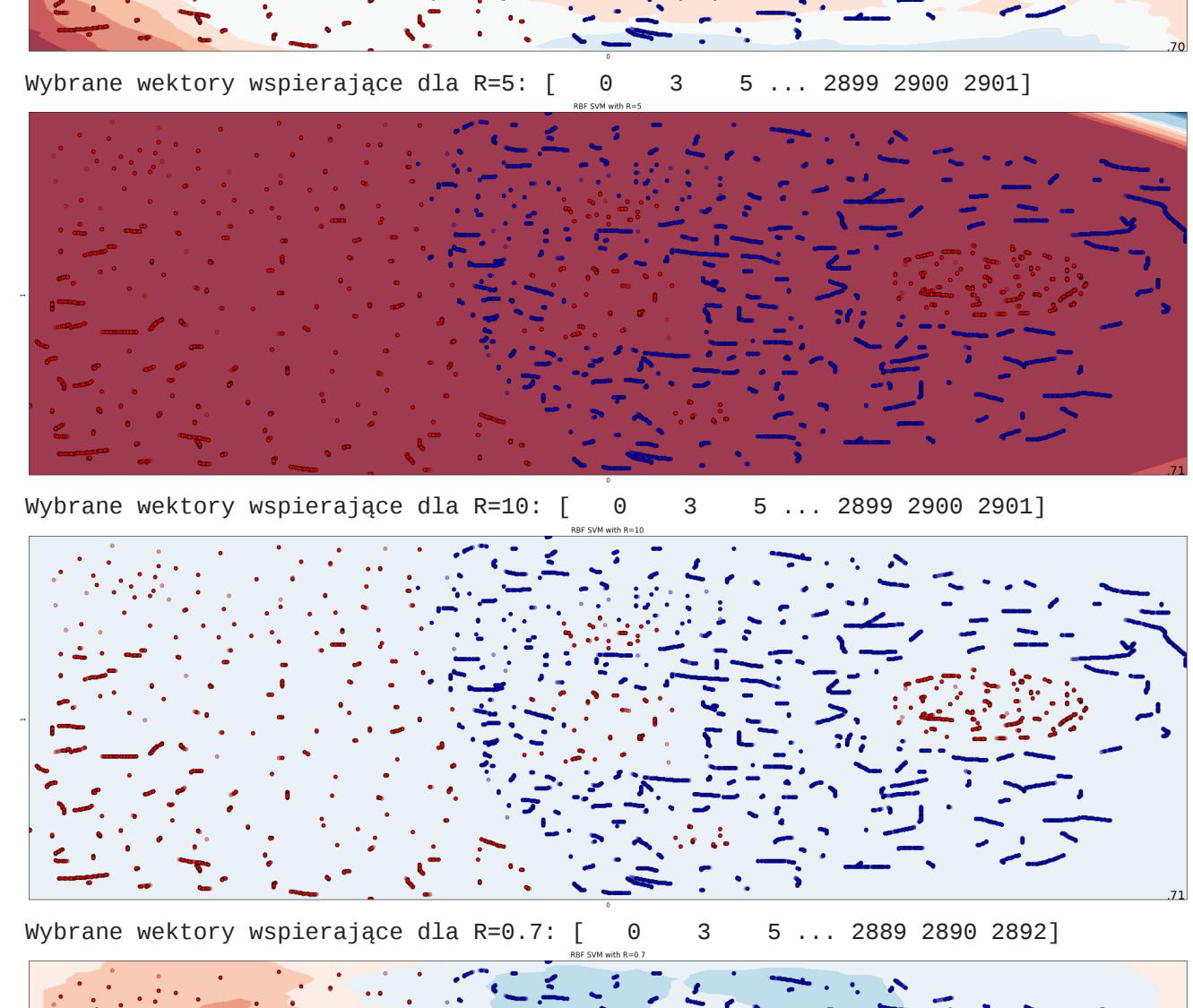
Wybrane wektory wspierające dla C=4: [0 7 8 ... 2893 2895 2900]

Odległość między marginesami klas: 1.823752890117034



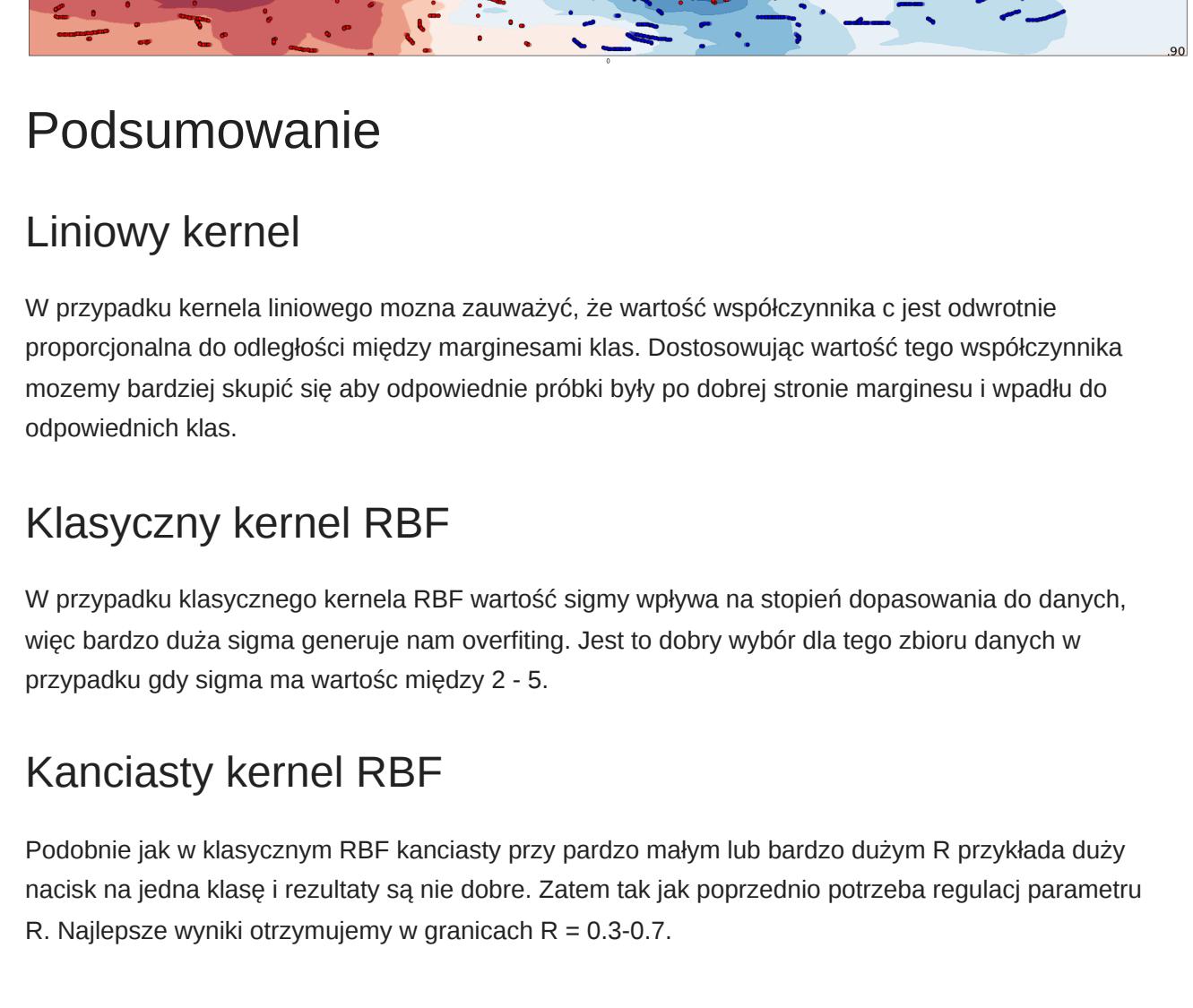
Wybrane wektory wspierające dla C=10: [0 7 8 ... 2893 2895 2900]

Odległość między marginesami klas: 1.8237528482804655



Wybrane wektory wspierające dla C=100: [0 7 8 ... 2893 2895 2900]

Odległość między marginesami klas: 1.8221826419807663



Podsumowanie

Liniowy kernel

W przypadku kernela liniowego można zauważać, że wartość współczynnika c jest odwrotnie proporcjonalna do odległości między marginesami klas. Dostosowując wartość tego współczynnika możemy bardziej skupić się aby odpowiednie próbki były po dobrej stronie marginesu i wpadły do odpowiednich klas.

Klasyczny kernel RBF

W przypadku klasycznego kernela RBF wartość sigmy wpływa na stopień dopasowania do danych, więc bardziej duża sigma generuje nam overfitting. Jest to dobry wybór dla tego zbioru danych w przypadku gdy sigma ma wartość między 2-5.

Kanciasty kernel RBF

Podobnie jak w klasycznym RBF kanciasty przy bardzo małym lub bardzo dużym R przykłada duży nacisk na jedną klasę i rezultaty są nie dobre. Zatem tak jak poprzednio potrzeba regulacji parametru R . Najlepsze wyniki otrzymujemy w granicach $R = 0.3-0.7$.