

PUM raport 3 - Maszyny wektorów wspierających

Piotr Zawiślan

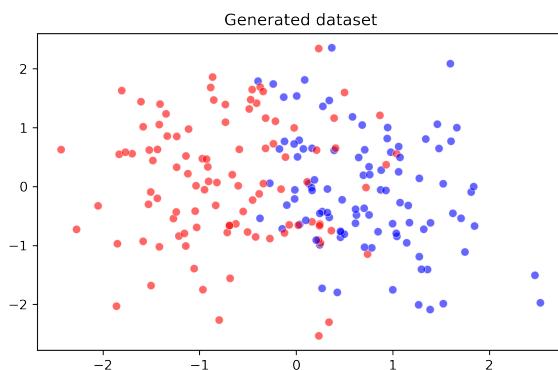
4 Maj 2022

1 Wykorzystane technologie

Kod napisałem w języku Python. Użyłem modułów Numpy, Scikit-Learn oraz Matplotlib/Seaborn.

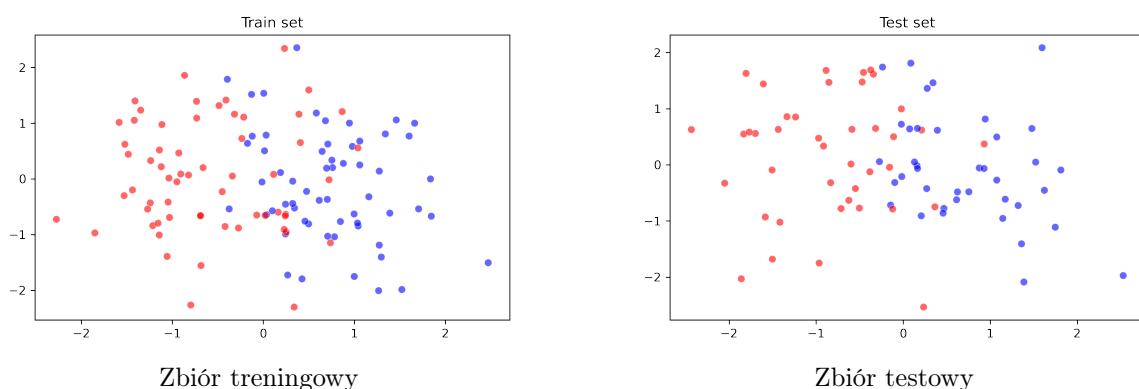
2 Dane do wykorzystania

Zbiór danych wygenerowałem przy użyciu funkcji `make_classification`.



Rysunek 1: Wygenerowany zbiór danych (punkty na płaszczyźnie należące do dwóch klas).

Podzieliłem go na zbiór treningowy i testowy używając klasy `StratifiedShuffleSplit`.



Rysunek 2: Zbiór treningowy i testowy (odpowiednio 60% i 40% całego zbioru).

3 Funkcja wizualizująca

Napisałem jedną funkcję wizualizującą, która na podstawie przekazanego obiektu klasyfikatora decyduje co ma wizualizować.

- Liniowy SVM, bądź SVM z jądrem liniowym: funkcję decyzyjną, wektory wspierające oraz margines.
- RBF Kernel: funkcję decyzyjną oraz wektory wspierające.
- Własny kernel: tylko funkcję decyzyjną.

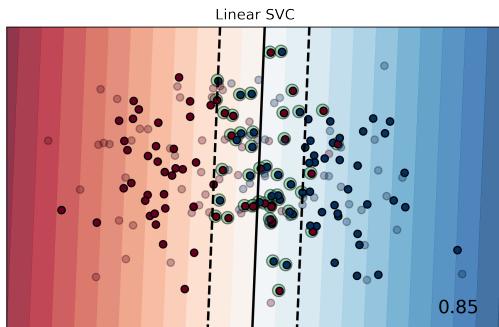
Funkcja rysuje też zbiór treningowy oraz testowy (elementy ze zbioru treningowego są bardziej wyraźne).

Funkcja wypisuje również szerokość marginesu (dla liniowej wersji) oraz zapisuje w prawym dolnym rogu obrazu dokładność modelu dla zbioru testowego.

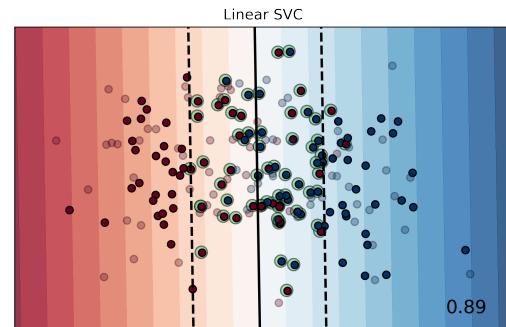
4 Testowanie SVM

4.1 Liniowy SVM

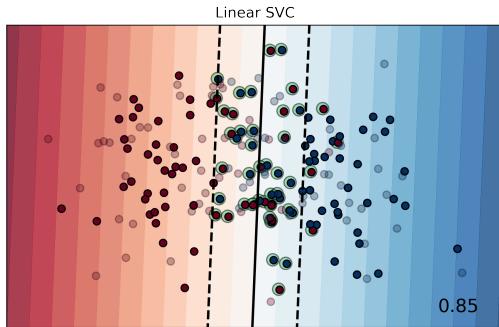
Liniowy SVM, bądź SVM z jądrem liniowym jest regularyzowany przez parametr C . Im C jest mniejsze tym mniej ważne to, że punkty przekraczają margines. Z tego wynika, że im mniejsze C tym margines będzie szerszy.



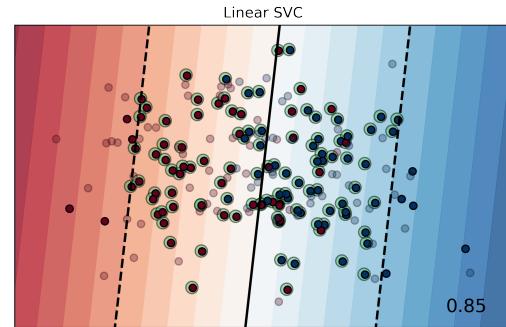
$C = 100$, szer. marginesu: 1.07



$C = 0.1$, szer. marginesu: 1.59



$C = 1000000$, szer. marginesu: 1.07



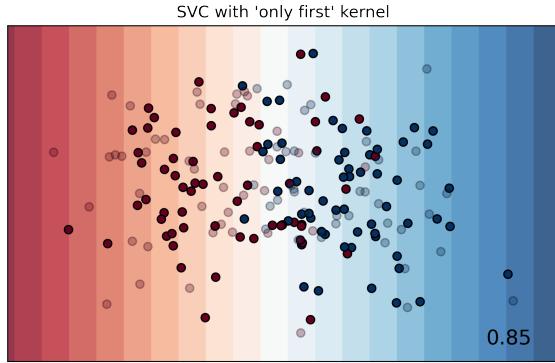
$C = 0.01$, szer. marginesu: 3.12

Rysunek 3: Liniowy SVM dla różnych wartości parametru C .

4.2 Kernel SVM

4.2.1 Iloczyn pierwszej składowej obu wektorów

Na tym wykresie widać, że funkcja decyzyjna zależy tylko od pierwszej składowej (poziomej).

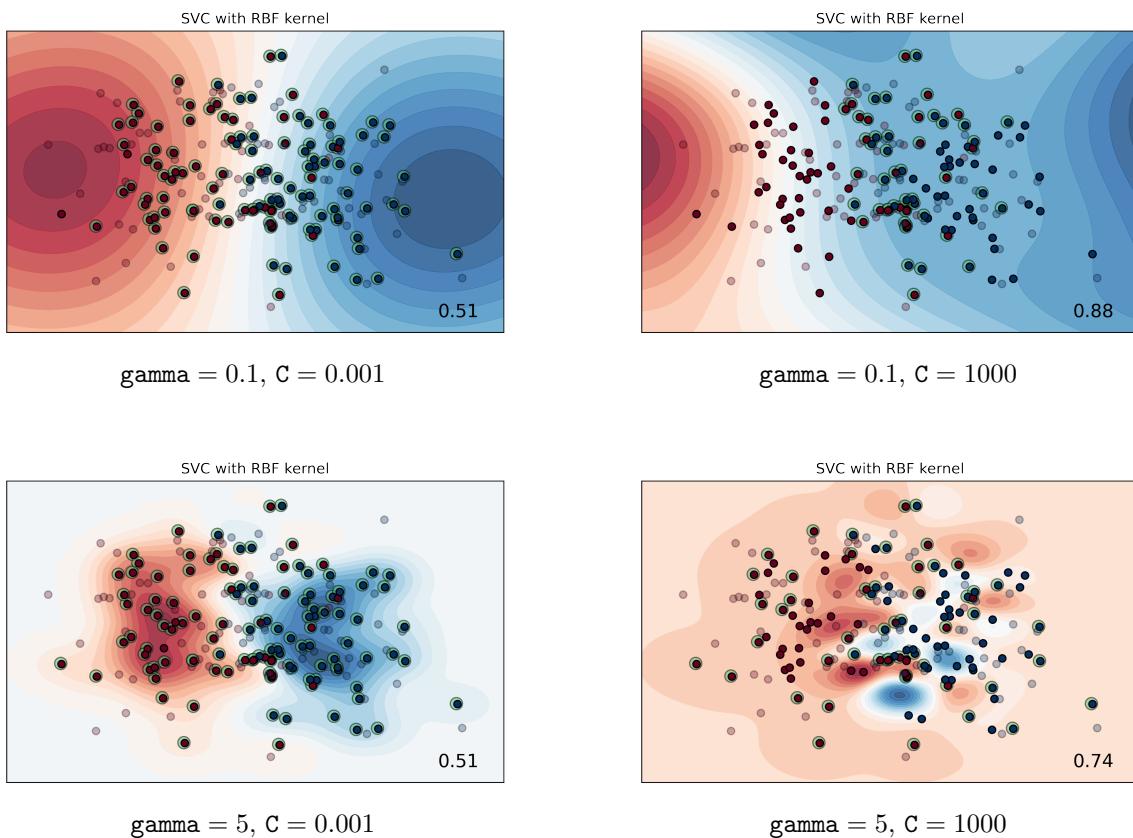


Rysunek 4: SVM z customowym jądrem liniowym pomijającym drugą składową.

4.2.2 Klasyczny kernel typu RBF

Kernel typu RBF ma dwa parametry: C - który już się pojawił oraz γ .

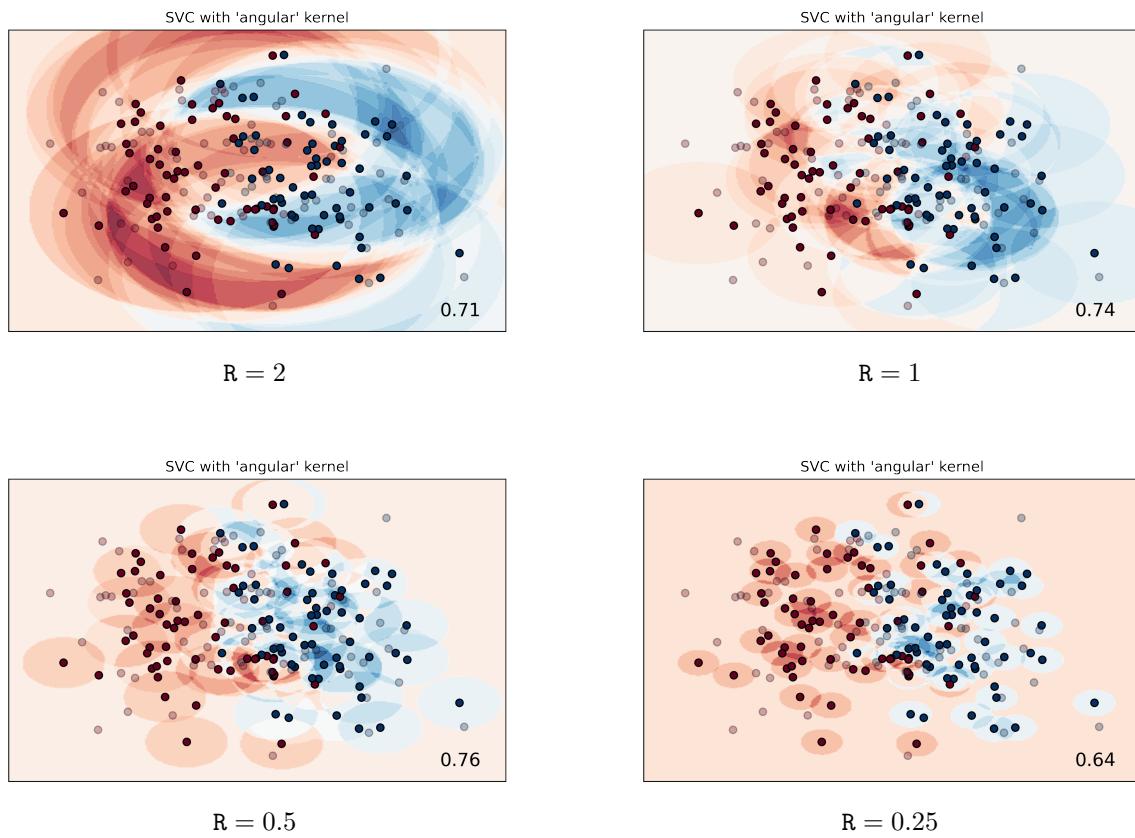
Im mniejsza γ tym większa regularyzacja (zamiast pojedynczych czerwonych/niebieskich plamek mamy dwa większe obszary).



Rysunek 5: SVM z jądrem RBF dla różnych wartości parametrów γ i C .

4.2.3 'Kanciasty' kernel typu RBF

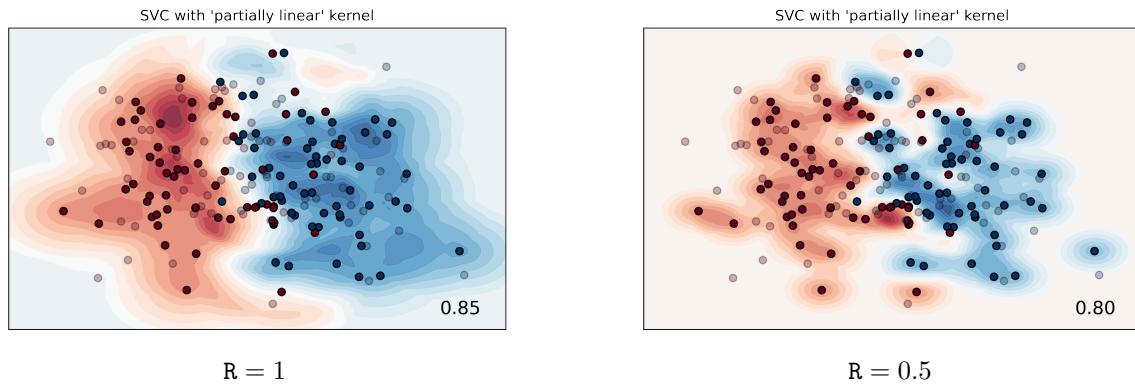
Im większe R tym większy obszar, na który ma wpływ pojedynczy przykład ze zbioru treningowego.



Rysunek 6: SVM z 'kanciastym' jądrem dla różnych wartości parametru R .

4.2.4 'Częściowo liniowy' kernel typu RBF

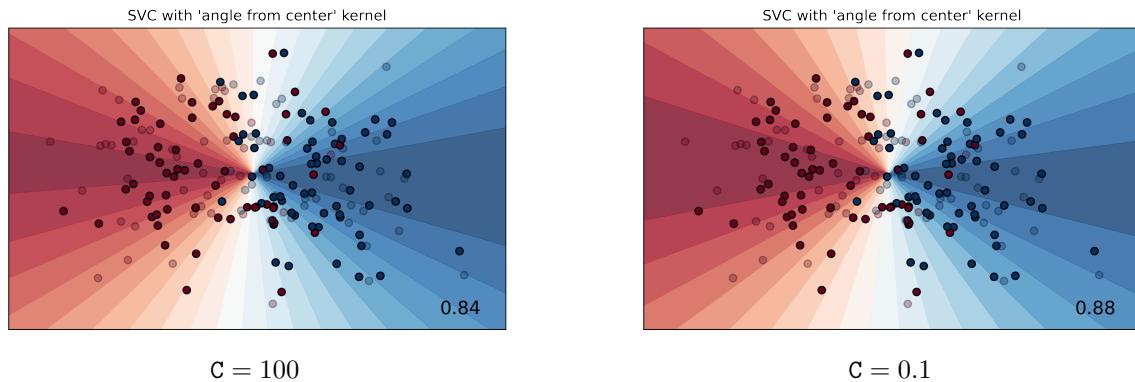
Jądro podobne do 'kanciastego', ale wpływ punktu ze zbioru treningowego na przewidywaną klasę stopniowo maleje wraz z odległością od tego punktu.



Rysunek 7: SVM z 'częściowo liniowym' jądrem dla różnych wartości parametru R .

4.2.5 Kernel zależny od kąta między wektorami (obliczanego od środka zbioru)

Na ostatnich wykresach widzimy, że większa regularyzacja w przypadku tego ciekawego jądra sprawia, że model się lepiej sprawdza na zbiorze testowym.



Rysunek 8: SVM z 'kątowym' jądreem dla różnych wartości parametru C .