

Algorytm SVM

Marcin Połosak

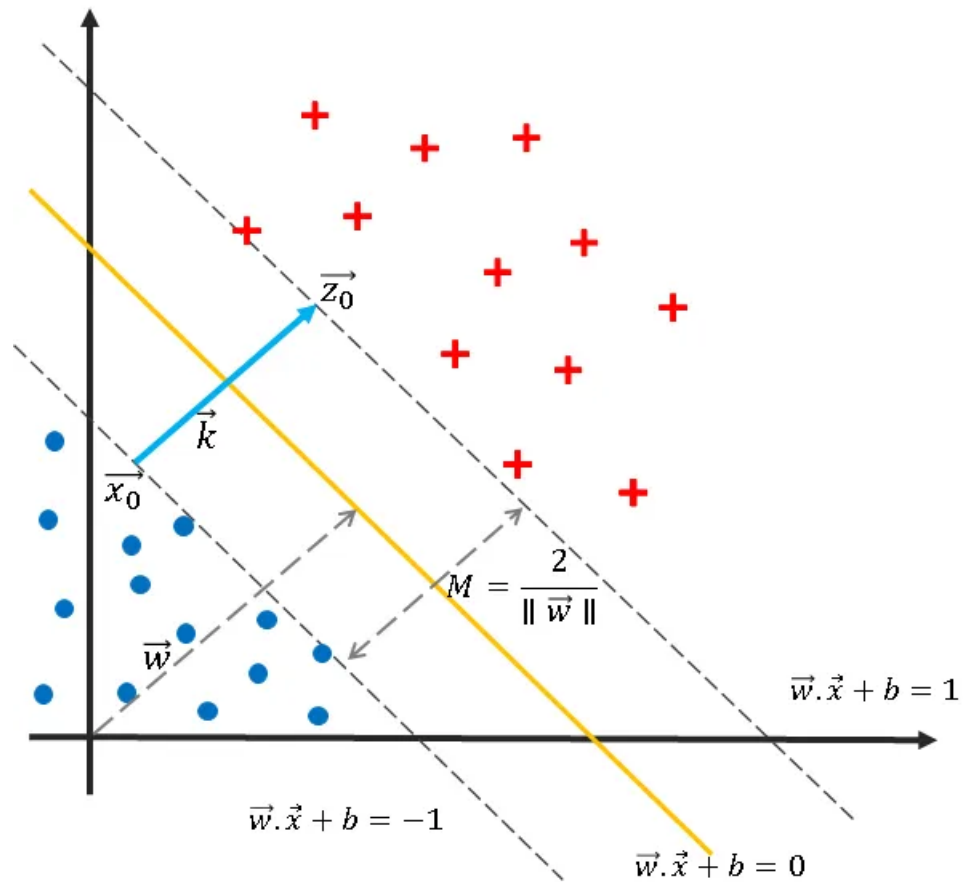
Grudzień 2023

1 Opis algorytmu

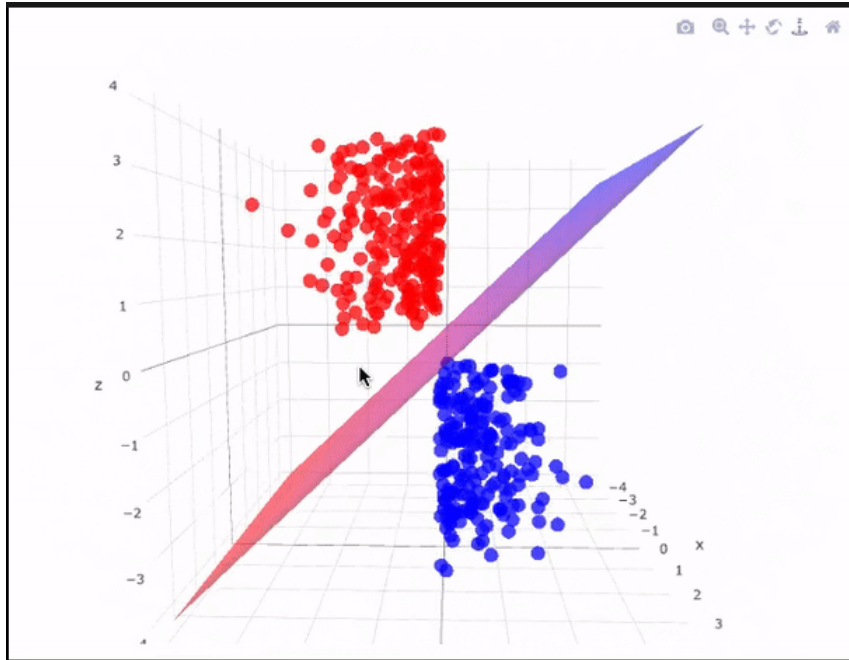
Algorytm SVM opiera się na kilku podstawowych założeniach zależnych od przyjętego wariantu. W swojej implementacji przyjąłem korzystanie z uczenia off-line wykorzystując algorytm przerobionego gradientu prostego dla optymalizacji. Kod opierał się na spełnieniu kolejnych punktów:

1. Ustalenie wag dla kolejnych parametrów za pomocą algorytmu gradientu prostego.
2. Wyznaczenie wyrazu wolnego
3. Dla podanego elementu testowego znalezienie odpowiedniej predykcji

Ogólną istotą problemu jest próba uzyskania jak najszerszej separacji liniowej dwóch zbiorów.



W przypadku zbiorów dwuwymiarowych problem ten można obrazować takim wykresem. Opieramy się na założeniu, że szerokość trasy wynosi 2, a jej wpasowanie w przerwę między zbiorami było uzyskane przez obrót płaszczyzny, na której się znajduje.



Celem algorytmu jest znalezienie funkcji decyzyjnej, której znak będzie określał predykcje do wartości Y dla zadanego X.

1.1 Algorytm gradientu prostego

W przypadku wykorzystanej przeze mnie postaci dualnej algorytmu celem wykorzystania algorytmu gradientu prostego będzie wyznaczenie odpowiedniej wartości parametru α , który posłuży do obliczenia wag stosując zależność:

$$w = \sum x_i y_i \alpha_i$$

Korzystając z mnożników Lagrange'a można zauważyć, że funkcja której maksimum należy wyznaczyć jest określona wzorem:

$$f(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Znajdująca się na końcu funkcja K to jądro. Można sobie wyobrazić sytuację, w której nie jest możliwe oddzielenie dwóch zbiorów linią prostą. Istnieje jednak założenie mówiące, że po odpowiednim przekształceniu elementów jest to możliwe w każdym przypadku. Istnieje wiele przykładów możliwych funkcji przekształceń. W swojej pracy skorzystam z dwóch:

1. Wielomianowe

2. Gausowskie

Dla zadanej funkcji znalazłem gradient, dzięki któremu będę w stanie zbliżyć się do odpowiedniej wartości parametru α :

$$\frac{\partial f(\alpha)}{\partial \alpha_i} = 1 - y_i \cdot \sum_{j=1}^N y_j K(x_i, x_j) \alpha_j$$

W przypadku korzystania z uczenia off-line możliwa jest implementacja zwykłej metody gradientu prostego bez konieczności implementacji gradientu stochastycznego.

Należy pamiętać o założeniu, mówiącym, że $0 \leq \alpha_i \leq C$, gdzie C jest przyjętą wartością i zabezpiecza przed nadmiernym wzrostem parametru.

1.2 Znajdywanie wyrazu wolnego

Parametr b wybieram korzystając z zależności, że jest to:

$$b = \text{avg}\{Y_i - w \cdot x_i\}$$

Złe dobranie wartości b może być jednym z głównych problemów niedokładności algorytmu, ponieważ decyduje on o późniejszej klasyfikacji próbki.

1.3 Funkcja predykcji

Nauczony model można wykorzystać teraz do predykcji wartości dla nowej próbki. Dla zadanego X_{test} korzystamy ze wzoru:

$$g(x_{test}) = w \cdot x_{test} + b$$

W algorytmie szukane jest rozdzielenie dwóch zbiorów. O wyniku predykcji decyduje znak powyższej funkcji rozdzielającej.

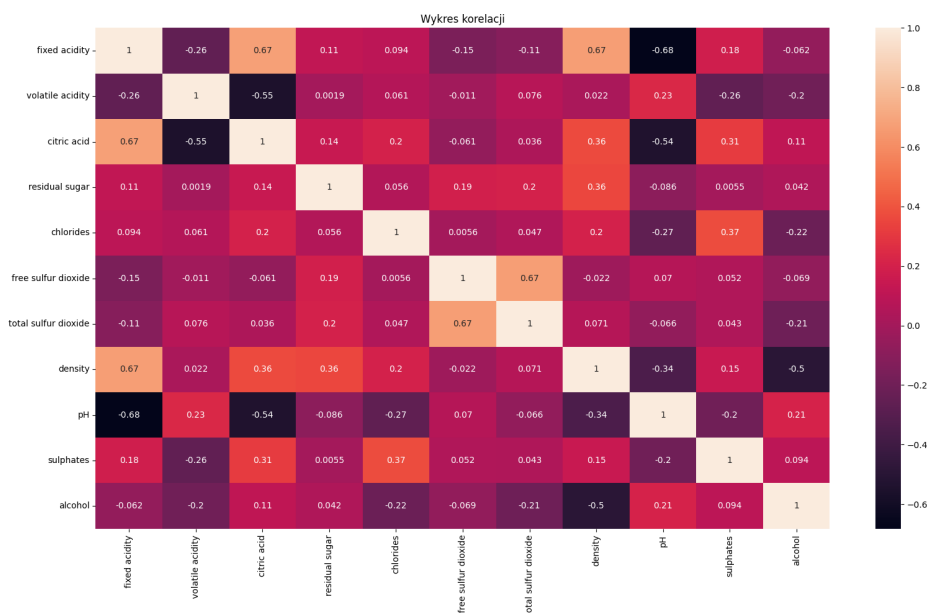
Jak widać, problem separacji jest dość skomplikowany, a na wartość uzyskanego wyniku wpływa wiele czynników. Właśnie dlatego tak trudne jest uzyskanie bardzo dokładnego algorytmu.

2 Analiza badanego zbioru

Algorytm SVM zostanie wykorzystany do analizy jakości wina na podstawie jego różnych parametrów.

	fixed acidity	volatile acidity	citric acid	...	pH	sulphates	alcohol
0	7.4	0.700	0.00	...	3.51	0.56	9.4
1	7.8	0.880	0.00	...	3.20	0.68	9.8
2	7.8	0.760	0.04	...	3.26	0.65	9.8
3	11.2	0.280	0.56	...	3.16	0.58	9.8
4	7.4	0.700	0.00	...	3.51	0.56	9.4
...
1594	6.2	0.600	0.08	...	3.45	0.58	10.5
1595	5.9	0.550	0.10	...	3.52	0.76	11.2
1596	6.3	0.510	0.13	...	3.42	0.75	11.0
1597	5.9	0.645	0.12	...	3.57	0.71	10.2
1598	6.0	0.310	0.47	...	3.39	0.66	11.0

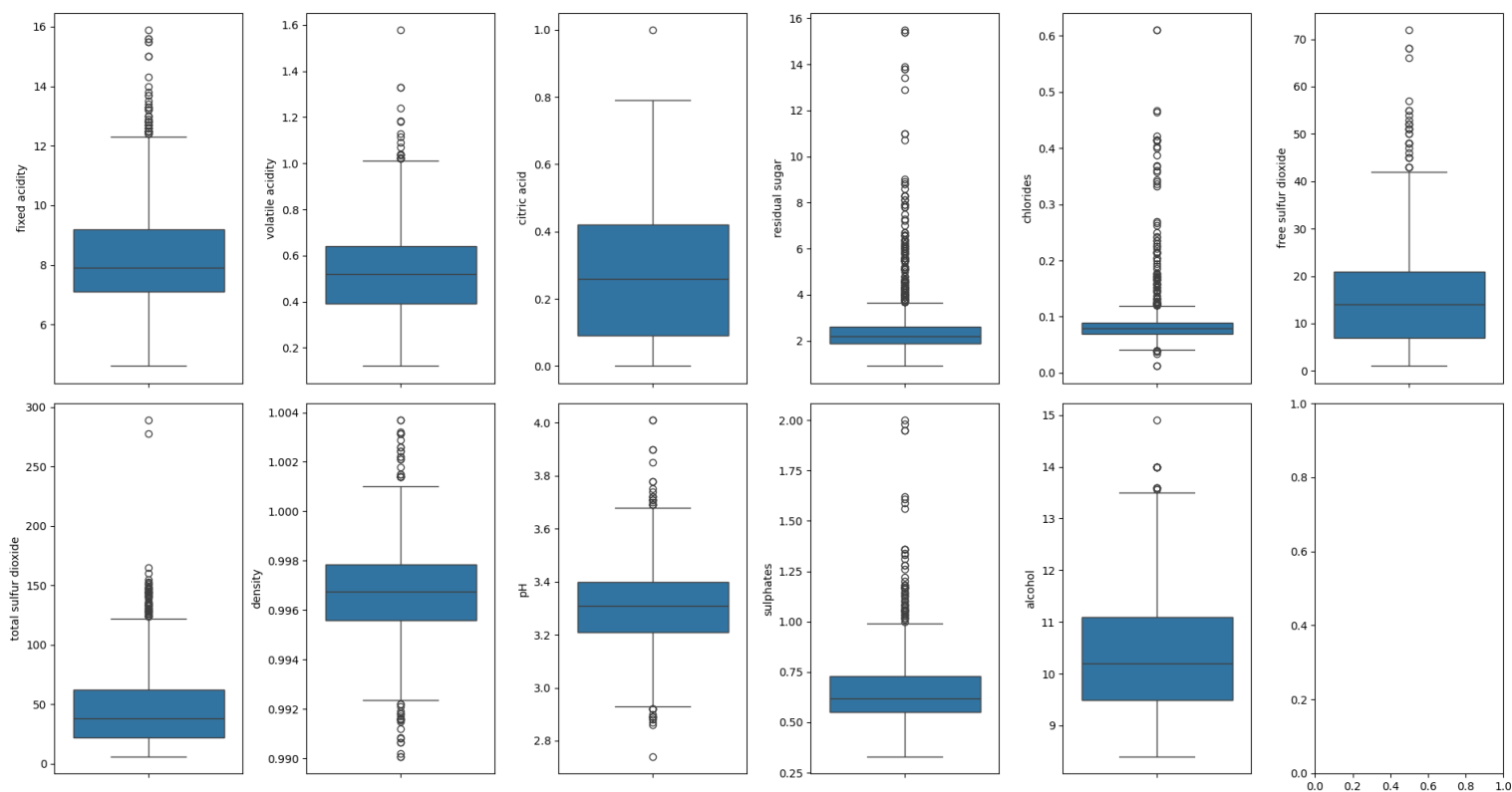
Oto przykład kilku z badanych parametrów. Można też pokazać ich wzajemną zależność na podstawie wykresu korelacji.



Badanie wykresu korelacji umożliwia stwierdzenie, że wzajemne powiązania parametrów istnieją, ale nie są na tyle znaczące, żeby można było zrezygnowanie z badania któreś ze zmiennych. W przypadku korelacji wynoszącej niemal 1/-1 oznaczałoby to, że zrezygnowanie z jednej z wartości nie wpłynie znacząco na poprawność działania modelu.



Kolejne sprawdzenie dotyczy wpływu poszczególnych czynników na jakość wina. Jak widać żadna ze zmiennych nie ma dominującego wpływu. Niektóre mają charakter marginalny, ale zdecydowałem o pozostawieniu ich w badanej próbie.



Ostatecznie przebadalem również wykresy pudełkowe każdego z parametrów. Duża ilość wartości odstających może być niepokojąca, ponieważ świadczy o nierównym rozkładzie wartości parametrów.

3 Eksperymenty

Testy algorytmu SVM wykonałem z wykorzystaniem dwóch funkcji jądrowych: wielomianowej oraz RBF. W ramach eksperymentu postanowiłem przebadąć poprawność działania implementacji dzieląc eksperymenty na trzy etapy.

3.1 Poszukiwanie najlepszego wyniku

W tej części eksperymentu starałem się modyfikować wartości hiperparametrów w celu znalezienia najlepszego rozwiązania. W tym celu dokonałem dyskretyzacji zmiennej objaśnianej (quality). Starałem się dokonać takiego podziału, dla którego oba zbiory będą miały podobną liczbę elementów. Zapewniło to możliwość uzyskania pewności, że uzyskany wynik nie jest tylko wynikiem szczęścia.

$$\begin{aligned} g(X) &= 1 & \text{dla} & & y(x) \geq 6 \\ g(X) &= -1 & \text{dla} & & y(x) < 6 \end{aligned}$$

Konieczne było również odpowiednie przeskalowanie danych w celu dopasowania ich przedziałów do optymalnych zasięgów. Przeprowadziłem testy, w których najlepsze wyniki uzyskiwałem dla starannie przyjętych parametrów. Osiągnięty wynik badałem mierząc "accuracy score", który porównywał uzyskane wyniki z prawdziwymi wartościami.

3.1.1 Wielomianowa

Funkcja wielomianowa opisana jest wzorem:

$$K(x, v) = (x \cdot v + a)^b$$

W przypadku mojej implementacji umożliwiam dobór parametrów a i b , które są odpowiednio parametrami kernel param1, kernel param2.

```
# SVM parameters
learning_rate = 0.02
lambda_param = 0.3
imax = 1000
limit = 100000
grad_error = 0.0003
kernel = "Polynomial"
kernel_param1 = 0.1
kernel_param2 = 4
```

```
Train sample:
1 in sample: 325
0 in sample: 314

Test result::
1 predictions: 580
0 prediction: 380
```

```
True values:
1 true value: 530
0 true value: 430
```

```
Accuracy_score: 0.7125
SKlearn accuracy score: 0.727
```

Dla przyjętych wartości udało mi się uzyskać accuracy score na poziomie 71.2%. Wynik ten stanowi dobrą dokładność. Jak widać, dla zadanej próbki wynik uzyskany przez SVM z biblioteki SKlearn jest tylko nieznacznie lepszy.

3.1.2 Gausowska

Jądro RBF ma określone jest wzorem:

$$K(x, v) = \frac{\|x - v\|^2}{\lambda}$$

Modyfikacji podlega tylko jeden parametr, który jest podawany na pierwszej wartości. Kernal param2 przyjmuje w tej części wartość domyślną (None).

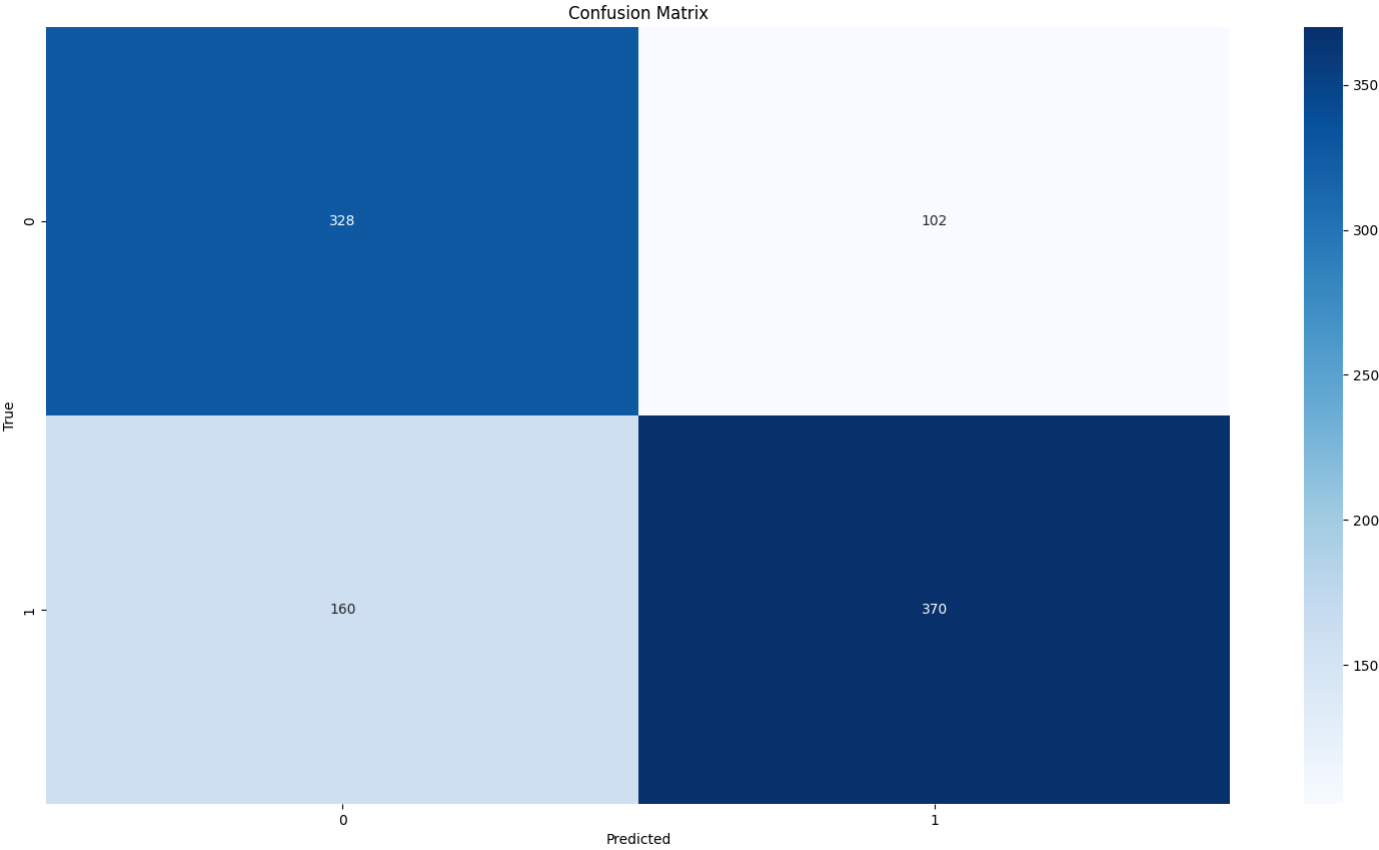
<pre># SVM parameters learning_rate = 0.02 lambda_param = 0.3 imax = 1000 limit = 100000 grad_error = 0.0003 kernel = "Gaussian" kernel_param1 = 1.0 kernel_param2 = 1</pre>	<pre>Train sample: 1 in sample: 325 0 in sample: 314 Test result:: 1 predictions: 531 0 prediction: 429 True values: 1 true value: 530 0 true value: 430 Accuracy_score: 0.703125 SKlearn accuracy score: 0.743</pre>
--	--

Również w tym przypadku uzyskany wynik wynosi ponad 70% i jest to wartość porównywalna z tą uzyskaną przy użyciu SKlearn z jądrem RBF.

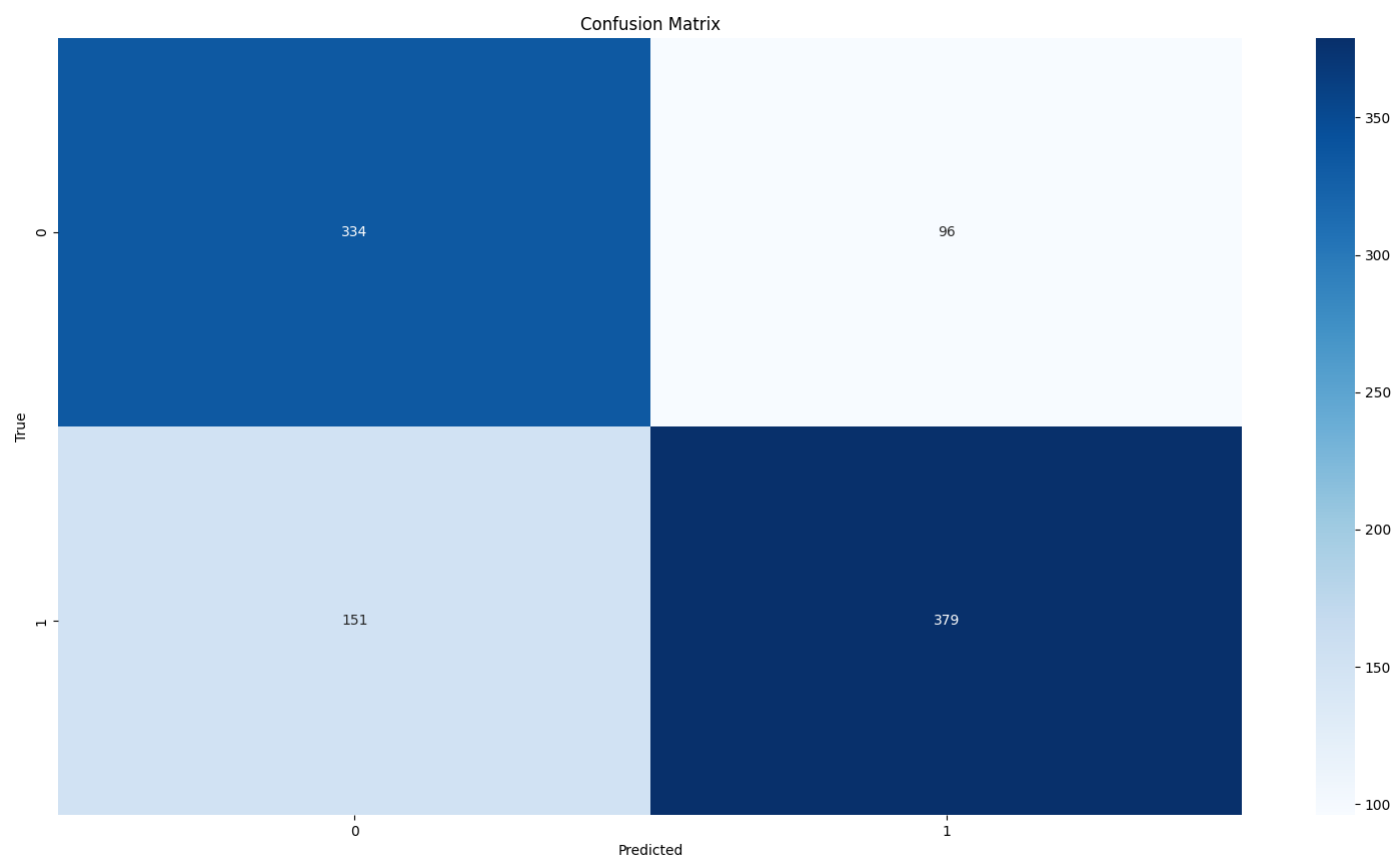
3.1.3 Wykresy rozstroju (Confusion)

Dla najlepszych moim zdaniem parametrów zbadalem również rozstrój, który pozwolił określić poprawność wyznaczonych podziałów.

1.Wielomianowa:



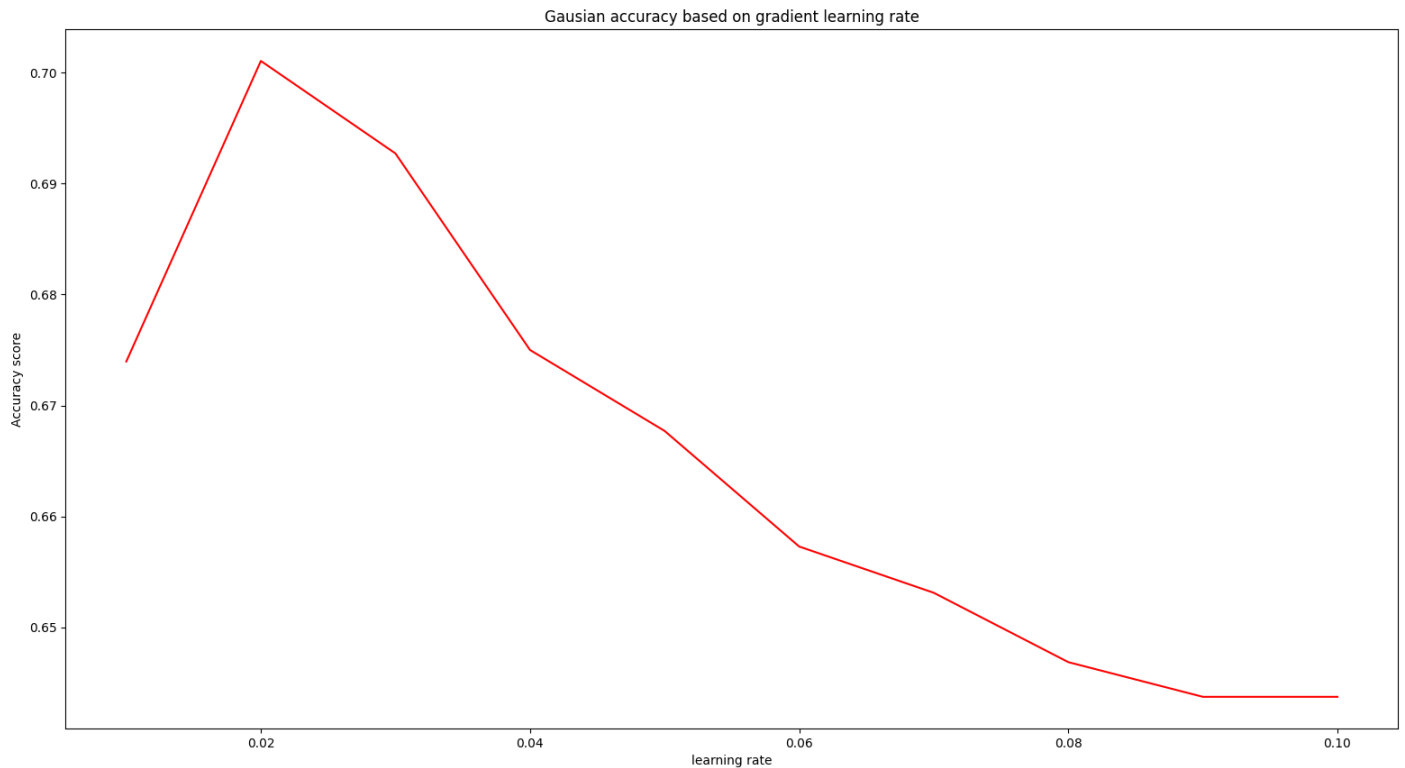
2.RBF:



Wykresy te udowadniają poprawność działania algorytmu. Jak widać w obu przypadkach algorytm nie ma znacznej tendencji do popełniania błędów w jedną ze stron. różnicę wynikają jedynie z lekkiej dysproporcji zbiorów separowanych.

3.2 Wpływ parametru "learning rate"

Dla jednej z funkcji jądrowych przebadalem jak na uzyskany wynik predykcji wpływała zmiana kroku gradientu prostego.



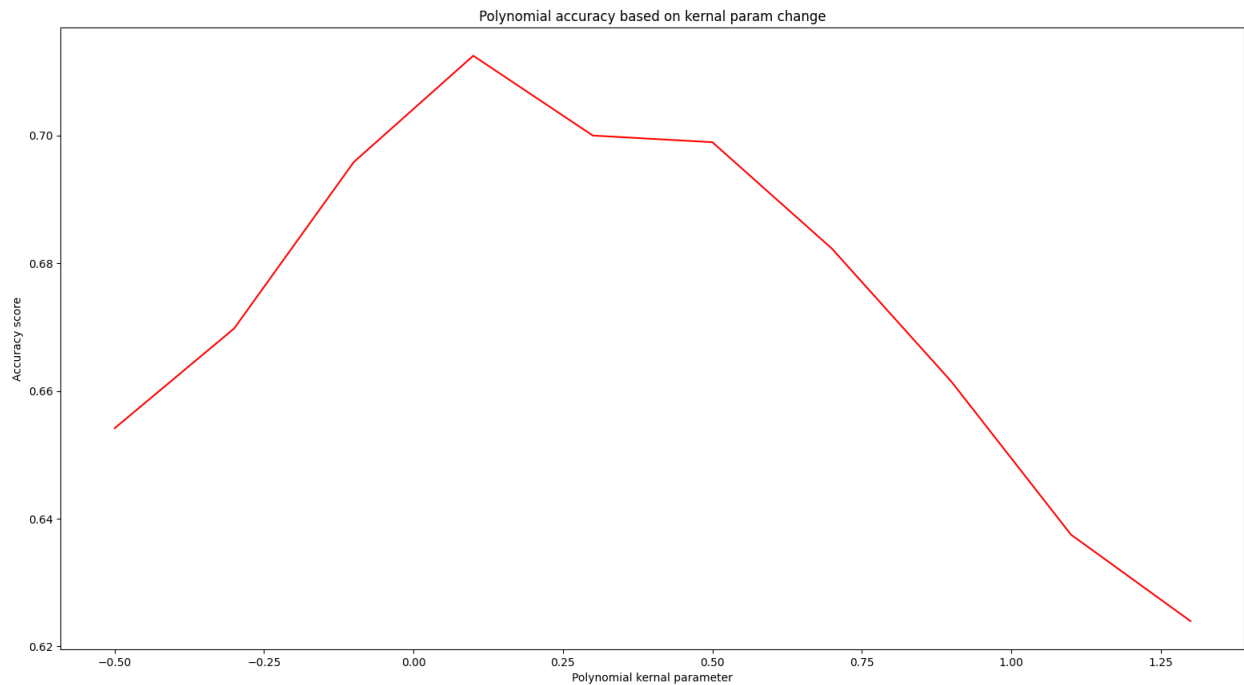
Można zauważyć, że stopniowe zwiększanie kroku powoduje odbieganie od wartości najlepszej, ponieważ układ zaczyna oscylować, a algorytm w zadanej maksymalnej ilości iteracji nie jest w stanie dość do tak dobrych wyników.

Oczywiście wartość za mała oddala nas od najlepszego rezultatu, ponieważ wówczas rośnie liczba potrzebnych do wykonania kroków.

3.3 Zmiana parametrów jądra

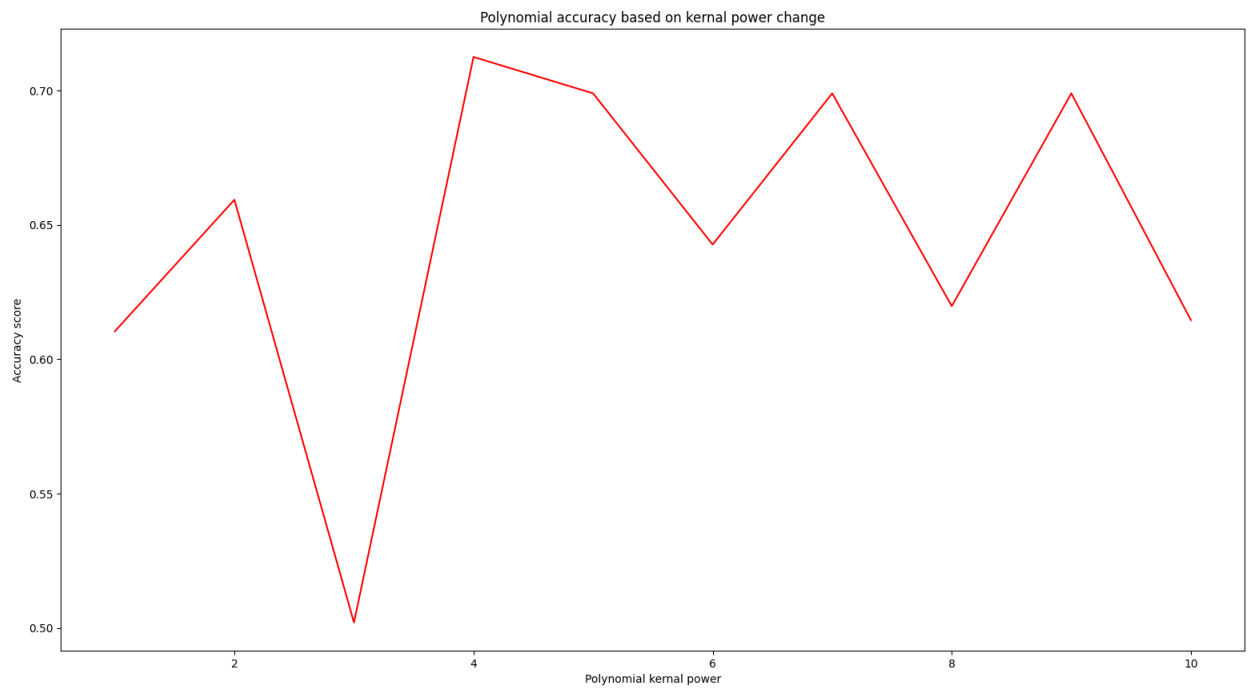
3.3.1 Wielomianowe

Przebadalem wpływ zmiany jednego z parametrów na wynik. Drugą wartość zostawiałem taką, jaką wcześniej przyjąłem za optymalną.



Jak widać, dla przyjętych wartości wykres jest dość płynny, jednakże można zauważyć, że zarówno przyjęcie za małych jak i za dużych wartości pierwszego parametru model traci na dokładności.

W moim modelu wartość parametru pierwszego powinna wynosić około 0.1.

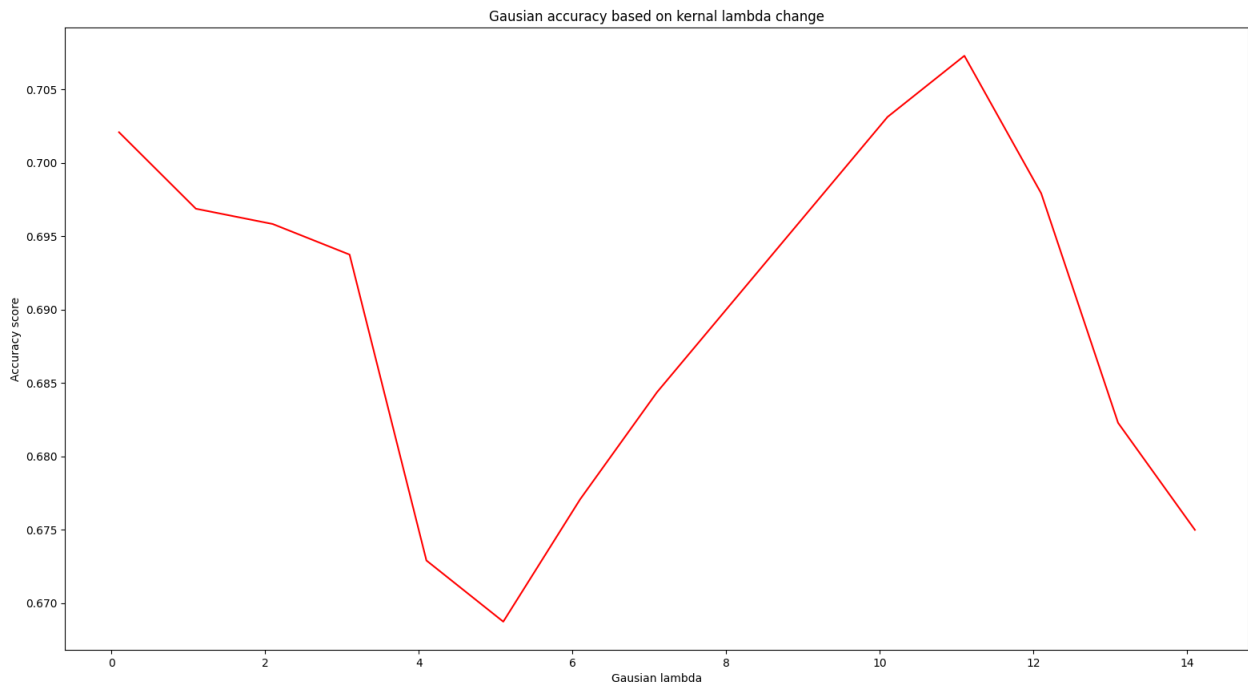


Znaczące zmiany widoczne są przy zmianie potęgi we wzorze. Dla kilku z przyjętych wartości rozwiązanie jest podobne, jednakże niektóre przekształcenia nie zapewniają poprawnej liniowej separowalności.

Dla moich danych zakładam, że najlepsze będzie użycie potęgi równej 4.

3.3.2 RBF

W tym przypadku do zbadania jest tylko parametr λ , który wpływa na uzyskane przekształcenie.



Wykres jest raczej dynamiczny. Wartość parametru λ ma dość duży wpływ na uzyskaną dokładność.

Przyjęcie wartości 11 wydaje się najrozsądniejszym rozwiązaniem.

4 Wnioski

Udało się uzyskać model, który cechuje się dość dużą dokładnością. Jest w stanie na podstawie badanej próbki nauczyć się klasyfikacji elementów. Zauważyłem, również, że dla przyjętych funkcji jądrowych zmiana każdego z parametrów ma wpływ na uzyskany wynik.

Dobór odpowiednich wartości hiperparametrów jest żmudny ale powinien zostać przeprowadzony starannie, ponieważ nieznaczne rozbieżności mają znaczący wpływ na wynik.