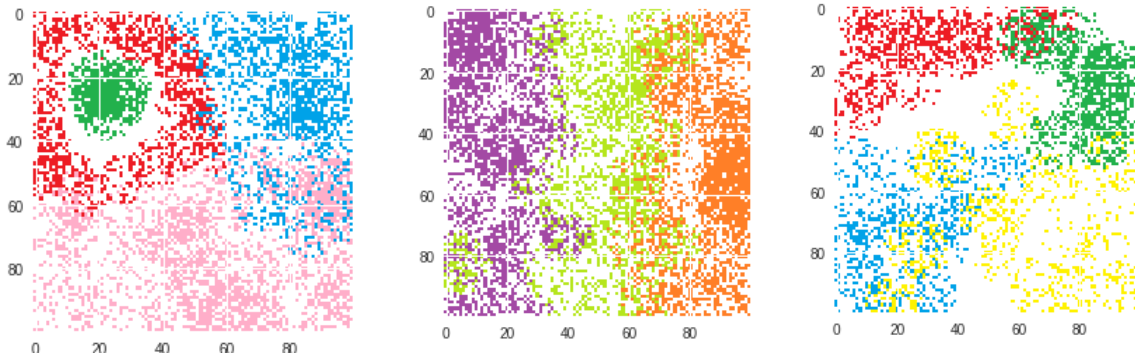


Raport – Metric Learning
Podstawy nauczania maszynowego
Wyk. Mateusz Woś

Do wykonania zadania użyłem danych wygenerowanych opisaną na zajęciach metodą „Paint”. Stworzyłem obrazy 100x100 pixeli i następnie za pomocą narzędzia sprej stworzyłem randomowe bazy stosując się do wszystkich zaleceń.

Dane wejściowe:



Dane zostały następnie zrzucone do formatu csv postaci z małym szumem :
(do każdego punktu dodałem randomowa wartość od 0 do 1)

Współrzędna x, współrzędna y, numer koloru, nazwa koloru.

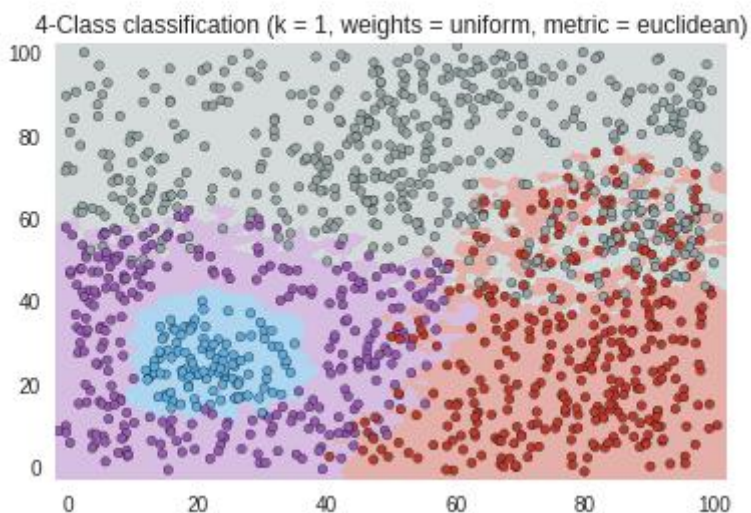
Dane normalizowałem do 10 podstawowych kolorów z pomocą biblioteki webcolors.

Do trenowania danych posłużyłem się już gotowym algorytmem knn z biblioteki scikit learn. Opierałem się na przykładach z dokumentacji.

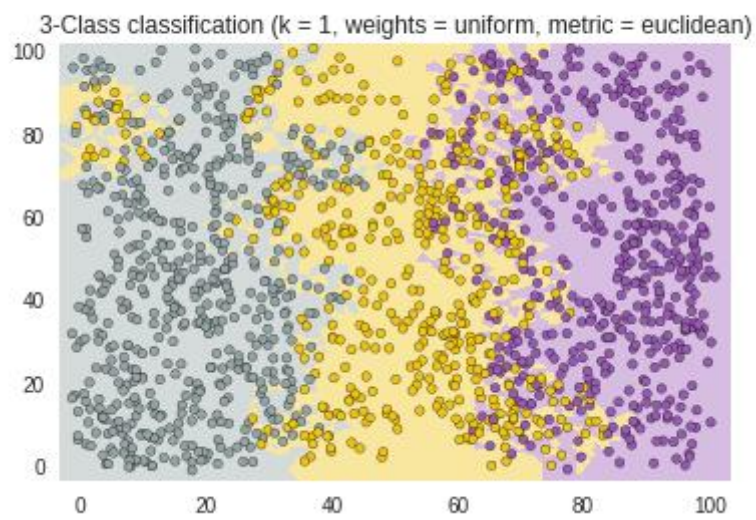
Wyniki otrzymane dla poszczególnych parametrów:

- knn with k=1, uniform weights and Euclidean metric

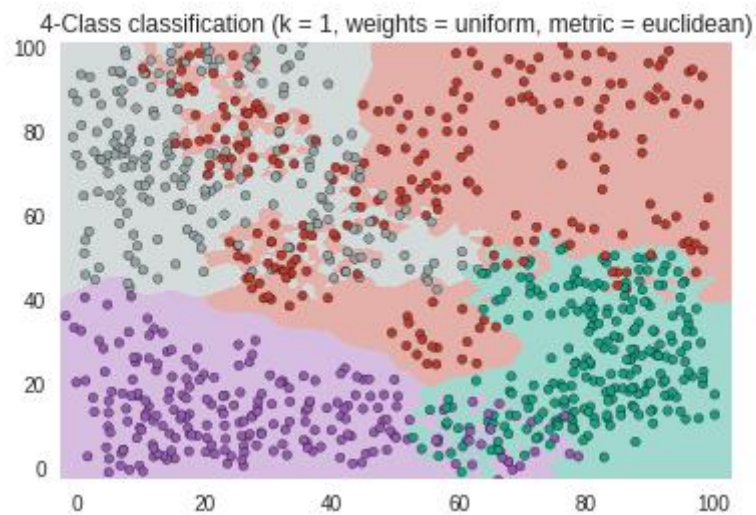
Accuracy: 0.8892380204241949



Accuracy: 0.8272425249169435

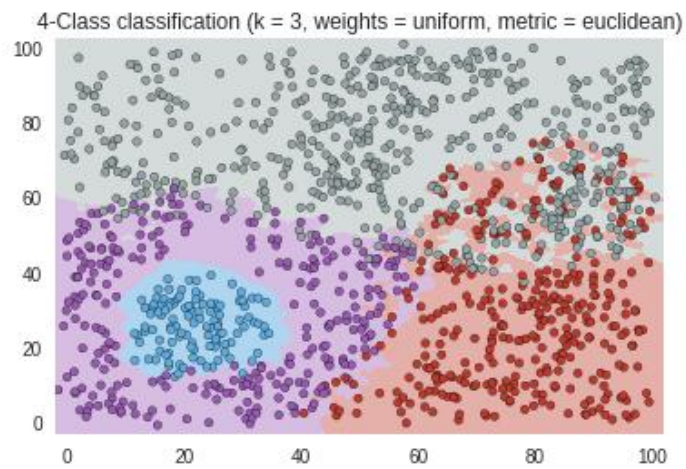


Accuracy: 0.8565169769989047

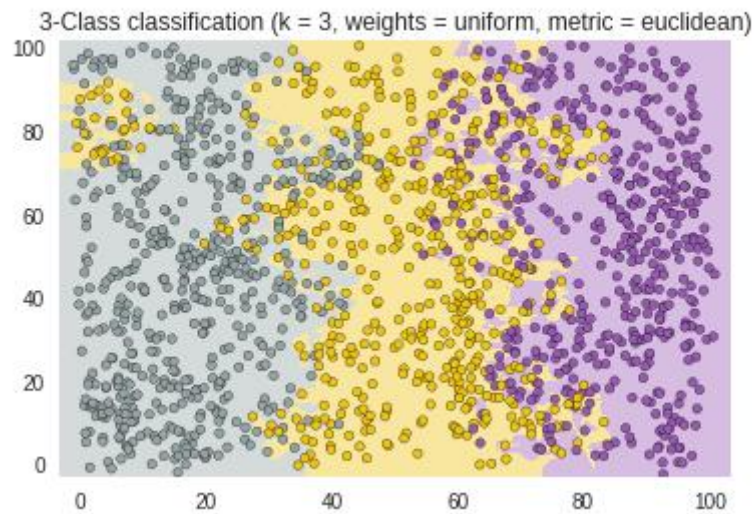


- kNN with $k=3$, uniform weights and Euclidean metric

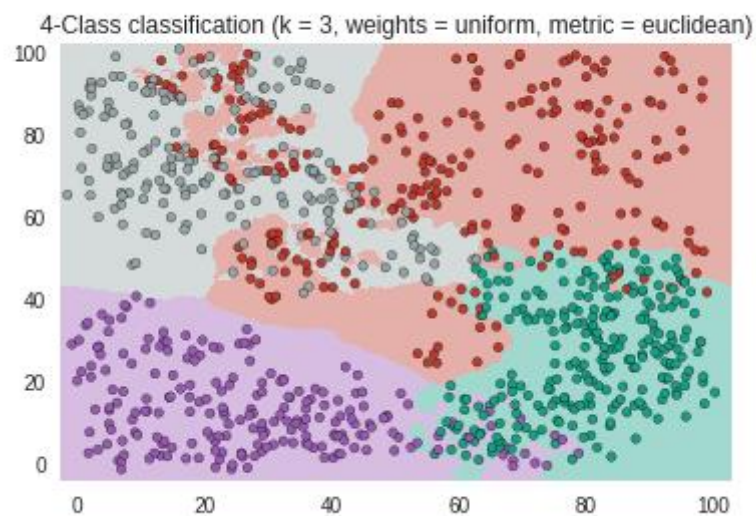
Accuracy: 0.8837391987431265



Accuracy: 0.837873754152824



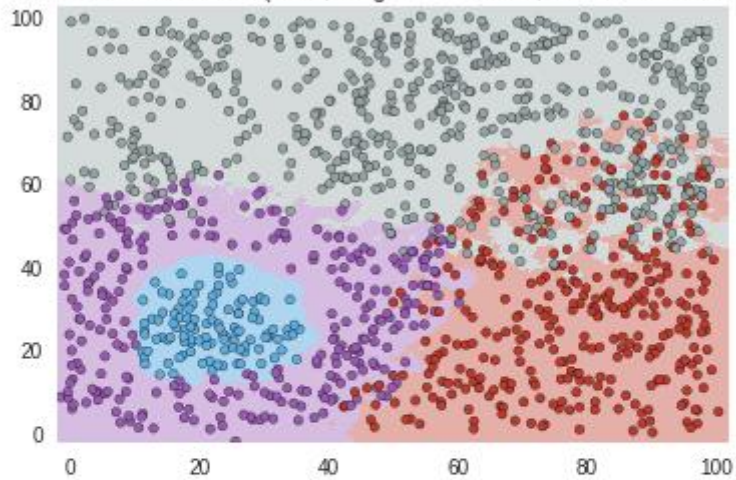
Accuracy: 0.8773274917853231



- kNN with $k=1$, distance weights and Euclidean metric

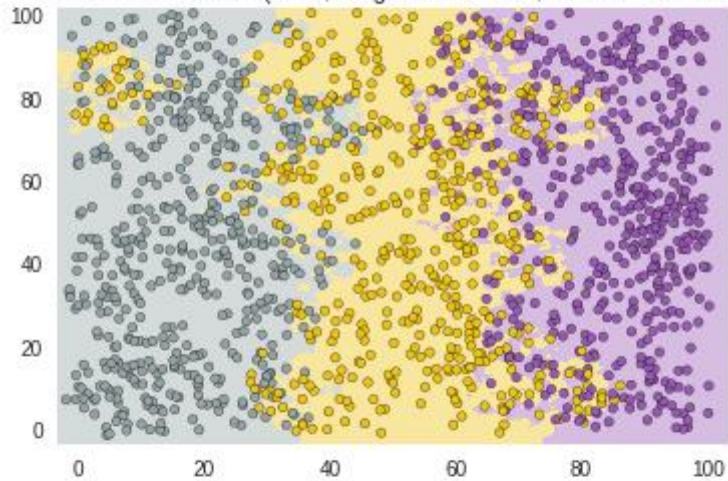
Accuracy: 0.8915946582875098

4-Class classification ($k = 3$, weights = distance, metric = euclidean)



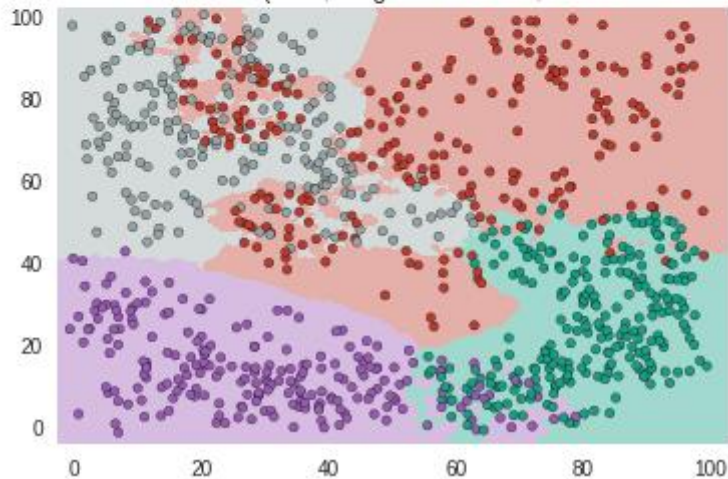
Accuracy: 0.8365448504983388

3-Class classification ($k = 3$, weights = distance, metric = euclidean)



Accuracy: 0.8543263964950711

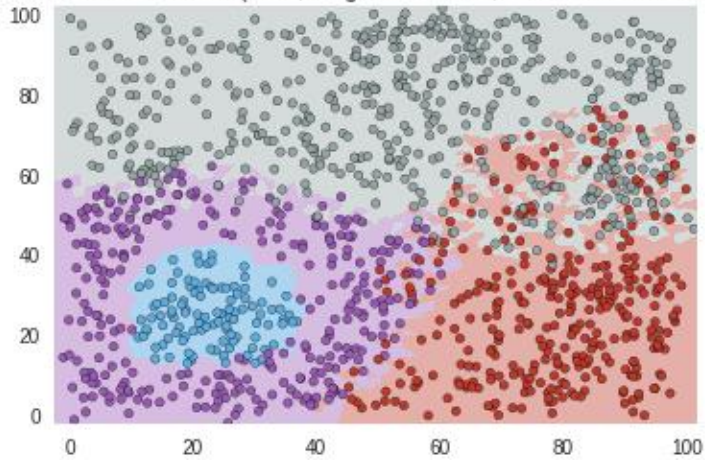
4-Class classification ($k = 3$, weights = distance, metric = euclidean)



- kNN with $k=1$, uniform weights and Mahalanobis metric

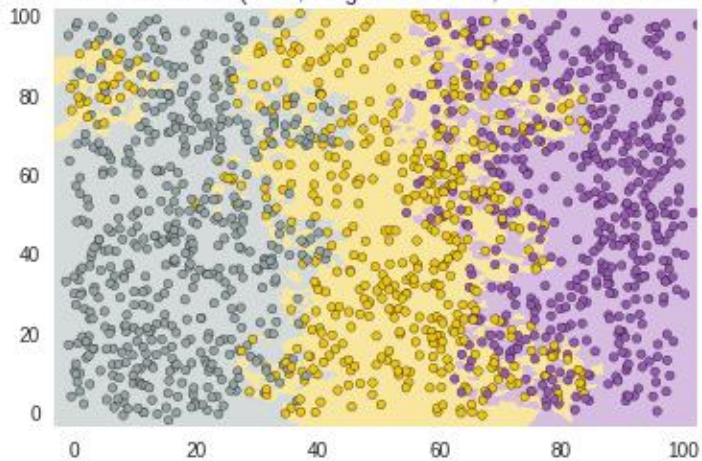
Accuracy: 0.9025923016496465

4-Class classification ($k = 1$, weights = uniform, metric = mahalanobis)



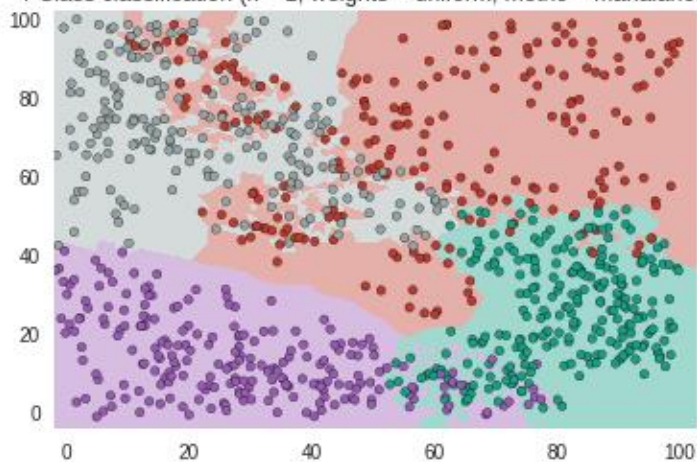
Accuracy: 0.8358803986710963

3-Class classification ($k = 1$, weights = uniform, metric = mahalanobis)



Accuracy: 0.859802847754655

4-Class classification ($k = 1$, weights = uniform, metric = mahalanobis)



Dopiero metryce Mahalanobis udało przebic sie 90% skuteczność na testowym data siecie 😊

- kNN with k=1, uniform weights and Mahalanobis metric depending on region

Niestety dla ponizszego przykladu nie udalo mi sie stworzyc odpowiednich wykresow. Nie potrafilem polaczyc wielu malych plotow w jeden duzy. Zabraklo i wiedzy jak i czasu.

Wyprintowałem natomiast skuteczność danych testowych na poszczególnych regionach.

```

Accuracy for segment (0,0): 1.0
Accuracy for segment (0,1): 1.0
Accuracy for segment (0,2): 0.7291666666666666
Accuracy for segment (0,3): 0.8888888888888888
Accuracy for segment (0,4): 1.0
Accuracy for segment (1,0): 0.9821428571428571
Accuracy for segment (1,1): 1.0
Accuracy for segment (1,2): 0.8611111111111112
Accuracy for segment (1,3): 0.9666666666666667
Accuracy for segment (1,4): 1.0
Accuracy for segment (2,0): 0.7704918032786885
Accuracy for segment (2,1): 0.7692307692307693
Accuracy for segment (2,2): 0.7045454545454546
Accuracy for segment (2,3): 1.0
Accuracy for segment (2,4): 1.0
Accuracy for segment (3,0): 1.0
Accuracy for segment (3,1): 0.9183673469387755
Accuracy for segment (3,2): 0.5319148936170213
Accuracy for segment (3,3): 0.6764705882352942
Accuracy for segment (3,4): 1.0
Accuracy for segment (4,0): 1.0
Accuracy for segment (4,1): 1.0
Accuracy for segment (4,2): 0.5949367088607594
Accuracy for segment (4,3): 0.6379310344827587
Accuracy for segment (4,4): 1.0
AVG ACCURACY FOR DATASET 1: 0.8812745915866286

Accuracy for segment (0,0): 1.0
Accuracy for segment (0,1): 1.0
Accuracy for segment (0,2): 1.0
Accuracy for segment (0,3): 0.8947368421052632
Accuracy for segment (0,4): 0.7424242424242424
Accuracy for segment (1,0): 0.6833333333333333
Accuracy for segment (1,1): 0.8679245283018868
Accuracy for segment (1,2): 0.8269230769230769
Accuracy for segment (1,3): 0.7377049180327869
Accuracy for segment (1,4): 0.7555555555555555
Accuracy for segment (2,0): 0.9772727272727273
Accuracy for segment (2,1): 0.9433962264150944
Accuracy for segment (2,2): 0.9565217391304348
Accuracy for segment (2,3): 0.6792452830188679
Accuracy for segment (2,4): 0.7837837837837838
Accuracy for segment (3,0): 0.6097560975609756
Accuracy for segment (3,1): 0.7580645161290323
Accuracy for segment (3,2): 0.6515151515151515
Accuracy for segment (3,3): 0.5662650602409639
Accuracy for segment (3,4): 0.7076923076923077
Accuracy for segment (4,0): 0.8793103448275862
Accuracy for segment (4,1): 1.0
Accuracy for segment (4,2): 1.0
Accuracy for segment (4,3): 0.9594594594594594
Accuracy for segment (4,4): 0.9767441860465116
AVG ACCURACY FOR DATASET 2: 0.8383051751907619

```

```

Accuracy for segment (0,0): 1.0
Accuracy for segment (0,1): 1.0
Accuracy for segment (0,2): 1.0
Accuracy for segment (0,3): 0.9736842105263158
Accuracy for segment (0,4): 0.75
Accuracy for segment (1,0): 1.0
Accuracy for segment (1,1): 1.0
Accuracy for segment (1,2): 0.6792452830188679
Accuracy for segment (1,3): 0.75
Accuracy for segment (1,4): 0.4594594594594595
Accuracy for segment (2,0): 0.851063829787234
Accuracy for segment (2,1): 1.0
Accuracy for segment (2,2): 0.7333333333333333
Accuracy for segment (2,3): 0.7592592592592593
Accuracy for segment (2,4): 1.0
Accuracy for segment (3,0): 0.6923076923076923
Accuracy for segment (3,1): 0.9310344827586207
Accuracy for segment (3,2): 0.8378378378378378
Accuracy for segment (3,3): 1.0
Accuracy for segment (3,4): 1.0
Accuracy for segment (4,0): 1.0
Accuracy for segment (4,1): 1.0
Accuracy for segment (4,2): 0.7777777777777778
Accuracy for segment (4,3): 1.0
Accuracy for segment (4,4): 1.0
AVG ACCURACY FOR DATASET 3: 0.887800126642656

```

Wnioski:

- W przypadku moich datasetów knn z większym parametrem k radził sobie o wiele lepiej. Jest to z pewnością spowodowane gęstością danych w obszarze a także ich rozrzuceniem. Wg moich testów (prob i błędów) na moich danych testowych najbardziej optymalne k jest w przedziale $[7,10]$. Zakładam, iż najlepiej byłoby to sprawdzić wyrysowując learning curves.
- Wybór pomiędzy głosowaniem ważonym i większościowe nie robiło większego znaczenia dla moich danych. Dane były zbyt gęsto rozsiane i metryka ważona nie pokazała tutaj swojego potencjału lub też wad.
- KNN z metryką Mahalanobisa radzi sobie delikatnie lepiej niż ten z metryką Euklidesową. KNN z metryką Mahalanobisa jako jedyny przekroczył 90% trafności na danych testowych.
- KNN z metryką Mahalanobisa działający na mniejszych obszarach radził sobie w wielu przypadkach idealnie, niektóre predykcje na danych testowych dawały 100% trafność. Niestety czasami zdarzały się tragiczne przypadki gdzie celność oscylowała w granicach 50%. W skutek czego średnia wszystkich wyników, nie była wcale lepsza niż w 4 przykładzie gdzie nie dzieliliśmy datasetu na mniejsze części. Obawiam się, iż może być to winna samego datasetu jak mojej implementacji.