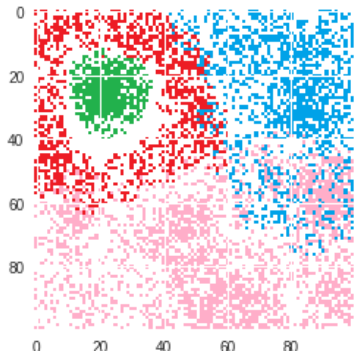
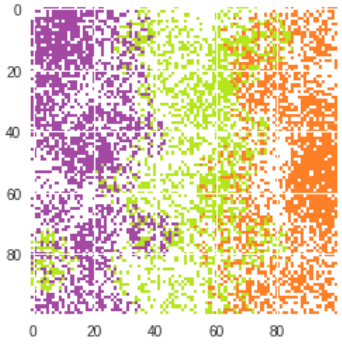
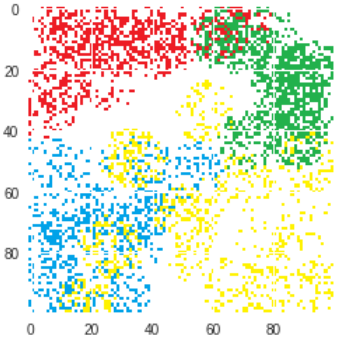
Raport – Condensed Nearest Neighbours
Podstawy nauczania maszynowego
Wyk. Mateusz Woś

Do powyższego zadani uzylem identycznych danych co w zadaniu z Metric learning.
Sposób wygenerowania danych opisałem w poprzednim zadaniu.

Dane wejściowe:

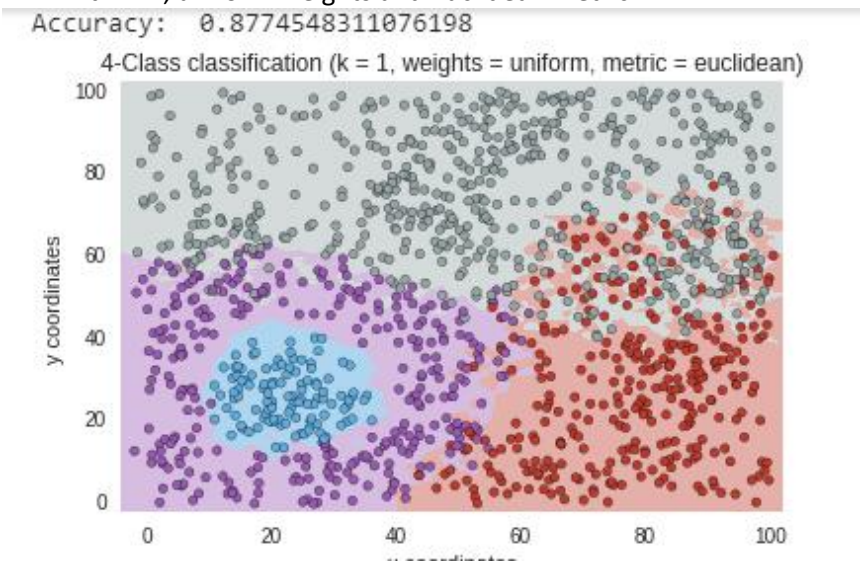




Dane zostały następnie tak jak w poprzednim zadaniu zrzucone do formatu csv postaci:
Wspolrzedna x, wspolrzedna y, numer koloru, nazwa koloru.

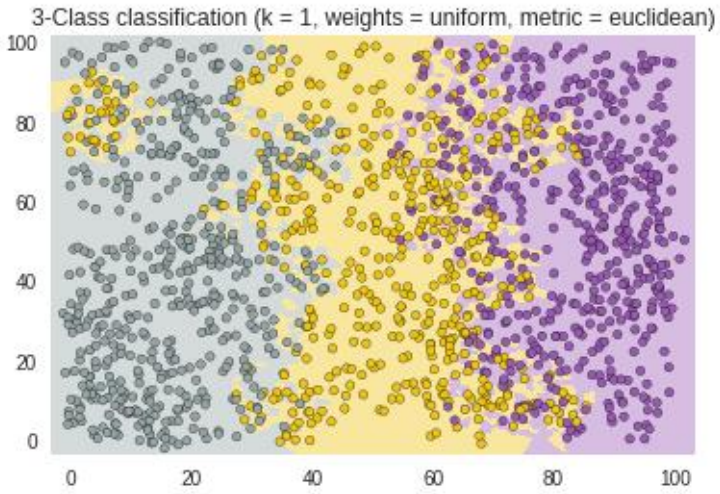
Kod do trenowania, filtrowania i wizualizacji w duzej mierze tez jest identyczny jak w zadaniu z Metric learning.
Biblioteka imblearn zapewnia gotowa implementacje under samplingu oparta o algorytm CNN, dzięki czemu oszczędziłem wiele czasu. Obiektowi CondensedNearestNeighbour z biblioteki nałożyłem tylko dane na których chciałem wykonać redukcje i powiedziałem mu ilu klas danych się spodziewamy.

Wyniki otrzymane dla poszczególnych parametrów:

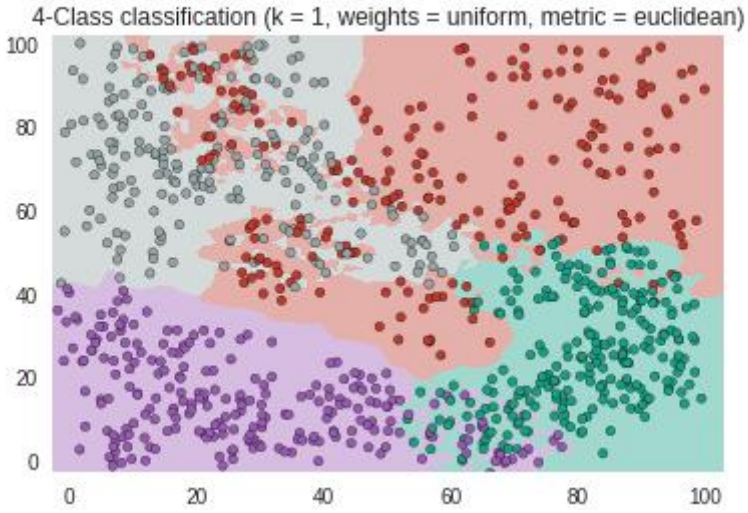
- kNN with k=1, uniform weights and Euclidean metric



```
Accuracy:  0.8385382059800665
```

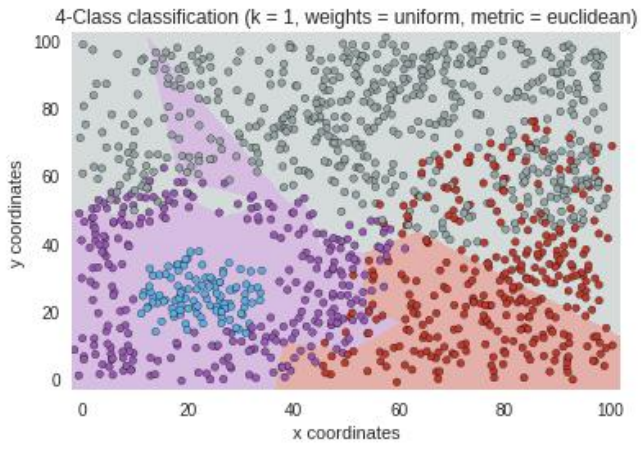



```
Accuracy:  0.872946330777656
```

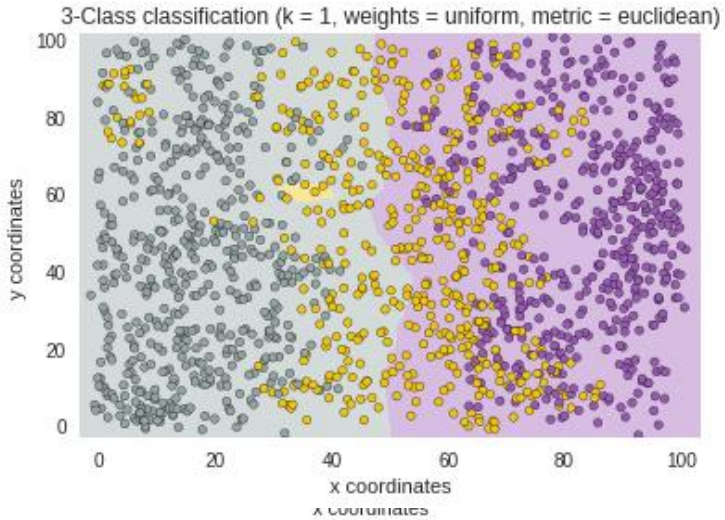


- CNN with k=1, uniform weights and Euclidean metric (randomly selecting samples in the condensation procedure)

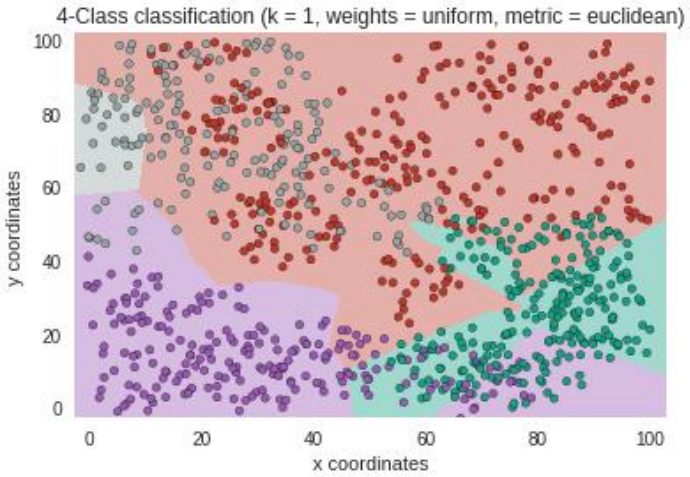
```
86 5940
Dataset ratio (reduced to whole dataset):  0.014478114478114479
Accuracy:  0.7085624509033779
```



```
910 7022
Dataset ratio (reduced to whole dataset):  0.12959270863001993
Accuracy:  0.6777408637873754
```




```
306 4258
Dataset ratio (reduced to whole dataset):  0.07186472522310944
Accuracy:  0.6757940854326396
```



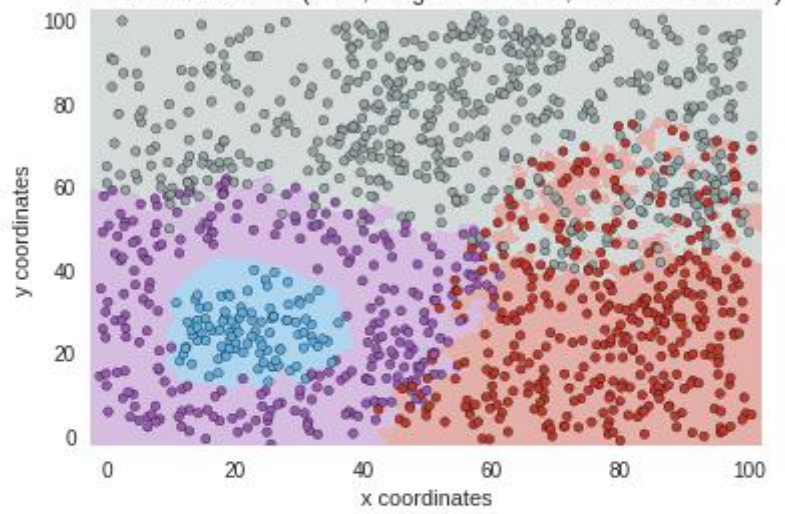
Widać tutaj ogromne zmniejszenie danych wejściowych do klasyfikatora. Przekłada się to jak widać na ogromne wygładzenie decision boundries. Najbardziej można to zaobserwować dla data setu nr 2.

- CNN with k=1, uniform weights and Euclidean metric (taking samples based on thier border ratio)

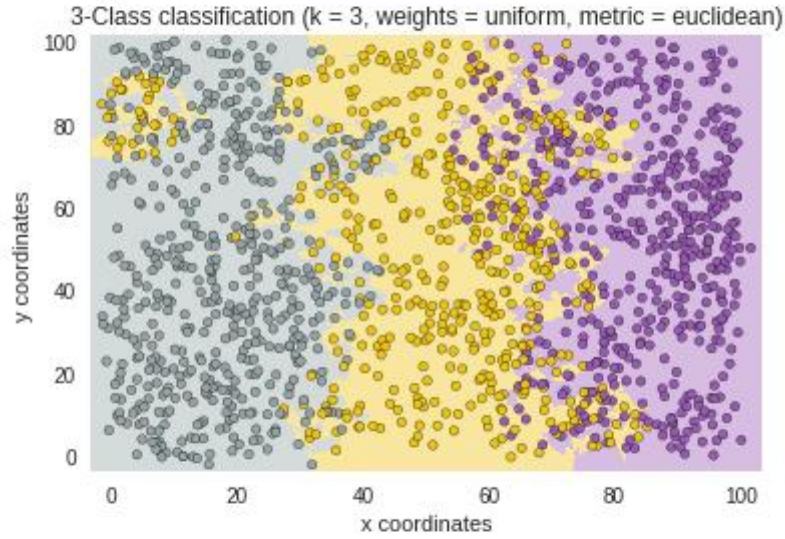
  Nie udało wykonać mi się tego popdunktu.

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

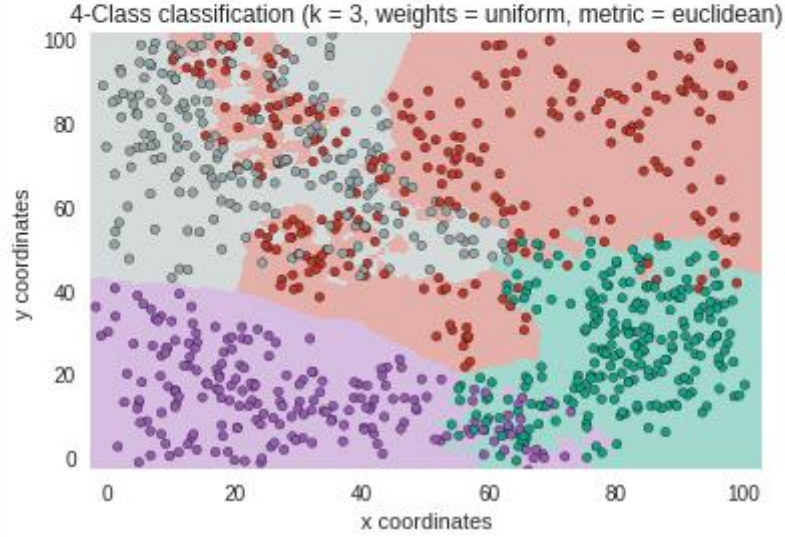
```
Accuracy:  0.8923802042419482
```




```
Accuracy:  0.8385382059800665
```

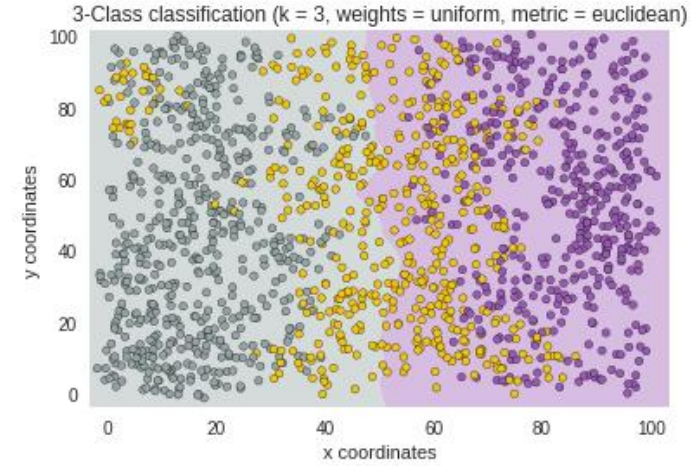



```
Accuracy:  0.8718510405257394
```

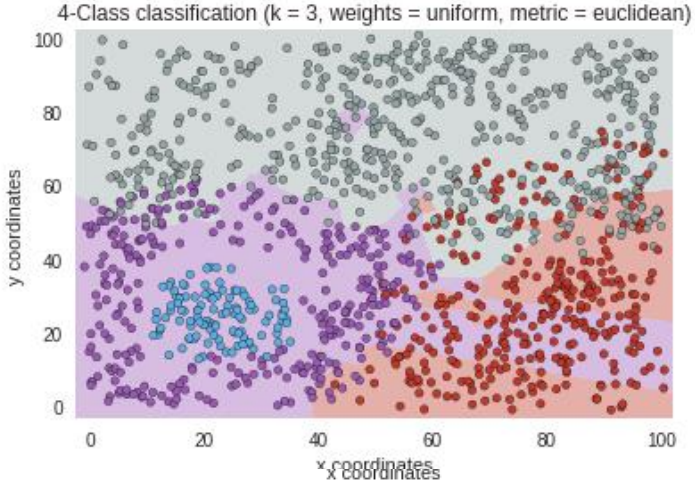


- CNN with k=3, uniform weights and Euklidesa metric

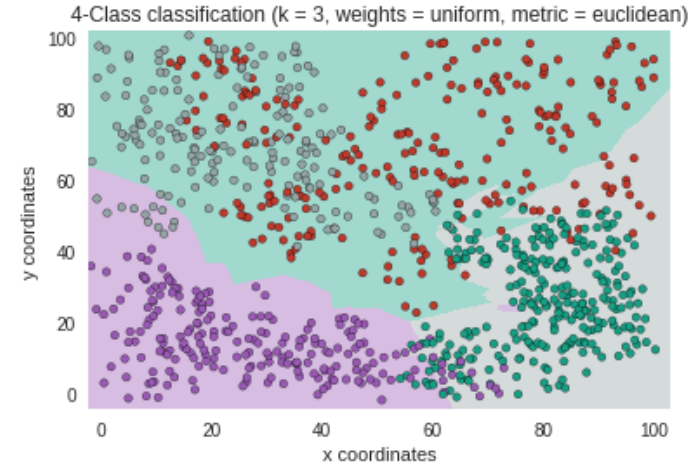
```
1048 7022
Dataset ratio (reduced to whole dataset):  0.1492452292794076
Accuracy:  0.6710963455149501
```




```
118 5940
Dataset ratio (reduced to whole dataset):  0.019865319865319864
Accuracy:  0.7054202670856246
```




```
376 4258
Dataset ratio (reduced to whole dataset):  0.08830436824800376
Accuracy:  0.7338444687842278
```



Dla k = 3 CNN zredukował nasze data sety do absurdalnie małych wartości. Decision boundry są teraz niesamowicie wygładzone. Nie sprawdzają się kompletnie dla datasetow.

Wnioski :

- Uzycie CNN na moich danych okazauje sie fatalnym pomyslem. CNN redukuje zbior treningowych do tego stopnia, ze dla k=3 mniejsze klasy danych nie sa brane w ogole pod uwage. CNN przydaje sie zapewne w przypadkach gdy klasy danych nie zachodza na siebie za bardzo.
- CNN z k=3 radzi sobie delikatnie lepiej niz z k=1, lecz ani troche nie dorownuja czystemu KNN.