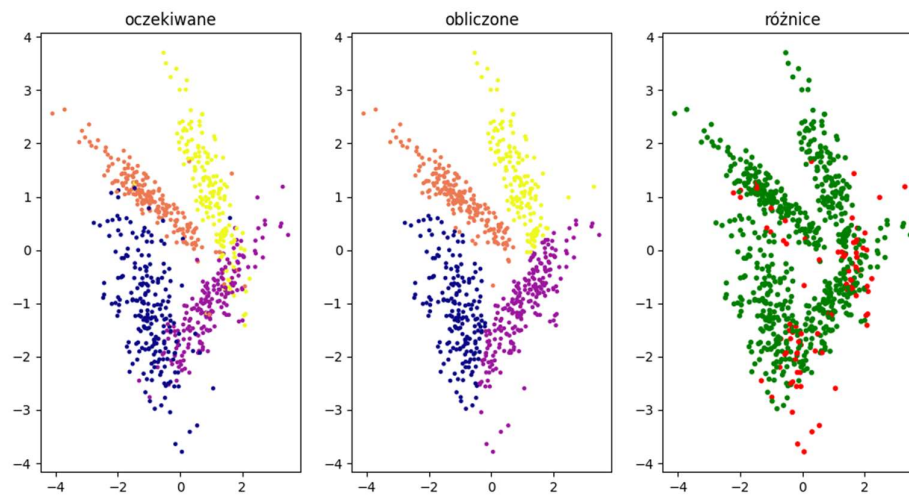
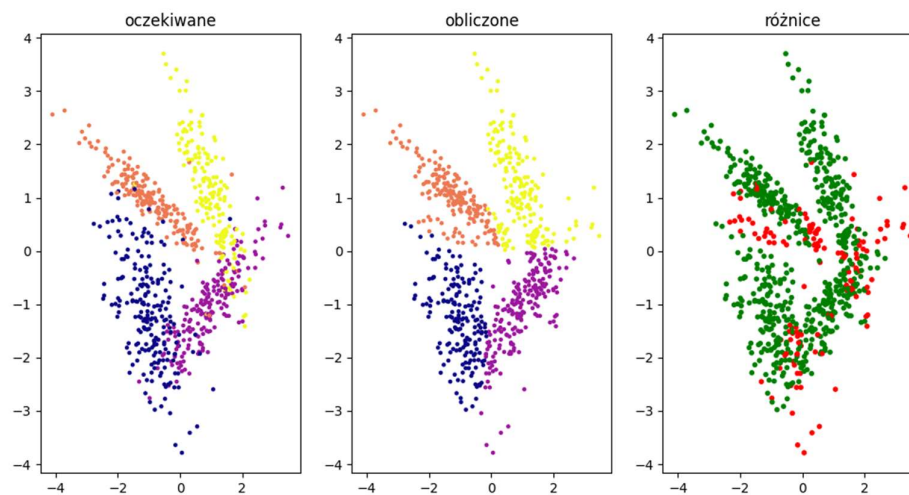


Wojciech Lidwin lw46577

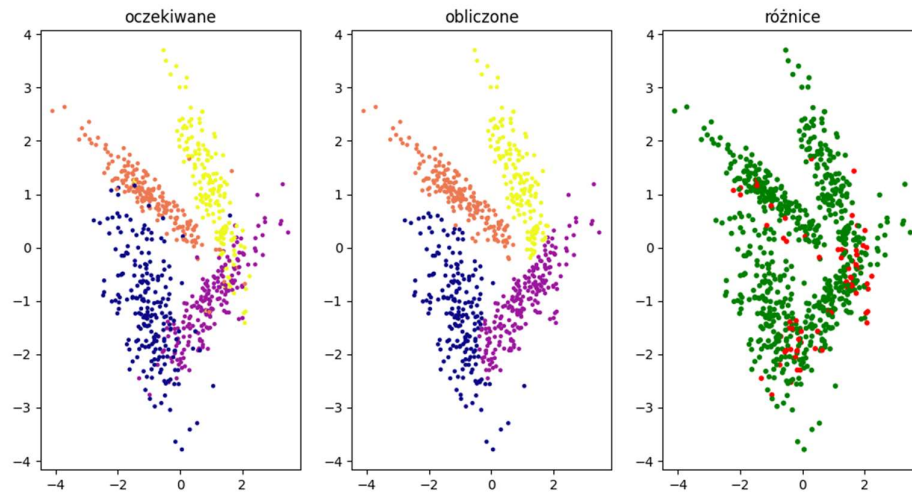
OneVsOneClassifier(), svm.SVC(kernel='linear', probability=True)



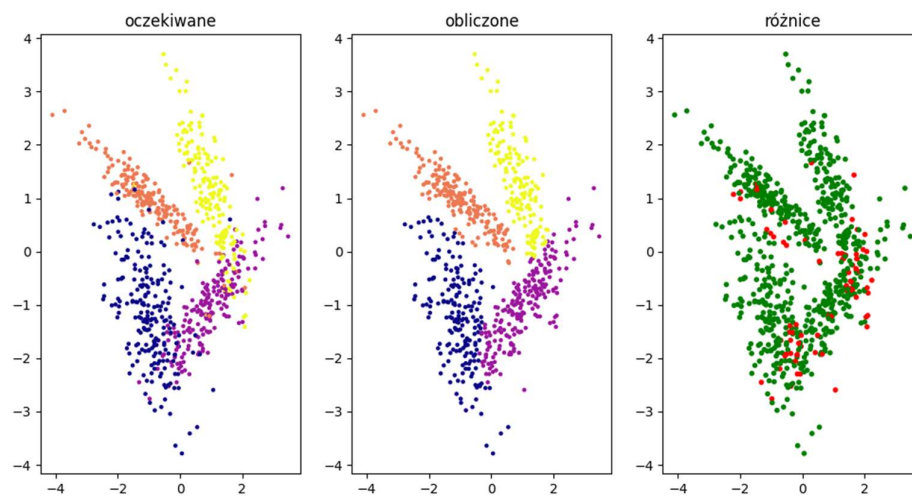
OneVsRestClassifier(), svm.SVC(kernel='linear', probability=True)



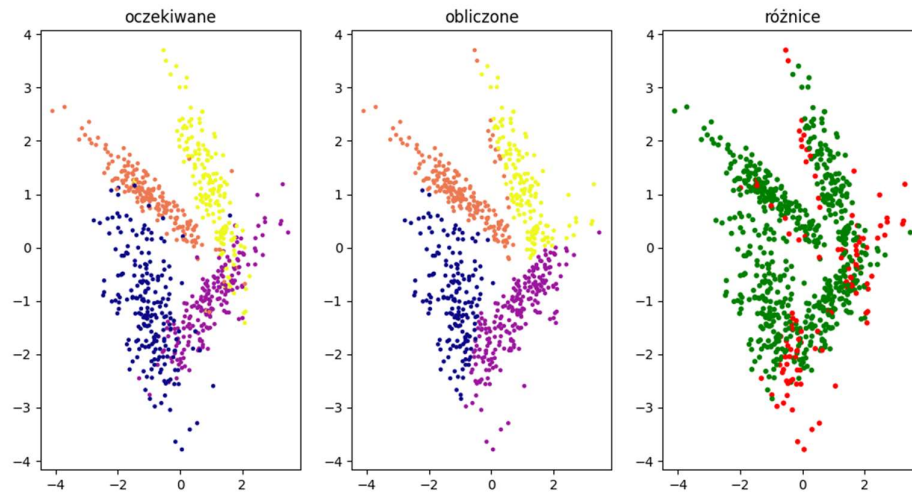
OneVsOneClassifier(), svm.SVC(kernel='rbf', probability=True)



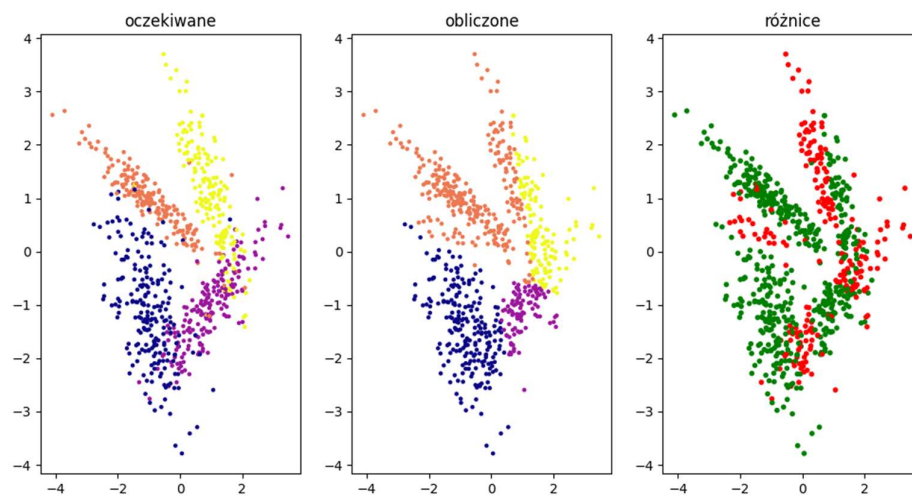
OneVsRestClassifier(), svm.SVC(kernel='rbf', probability=True)



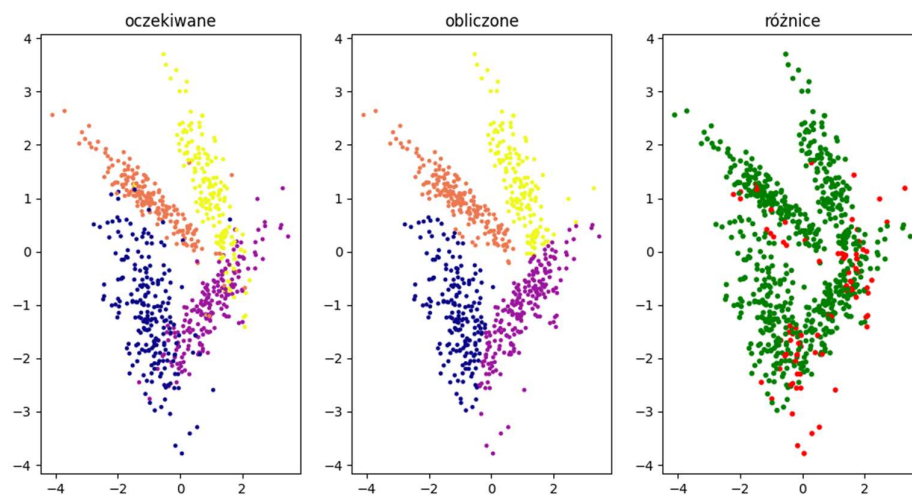
OneVsOneClassifier(), LogisticRegression()



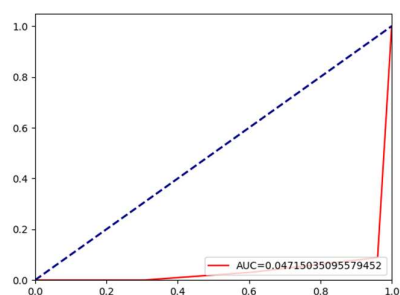
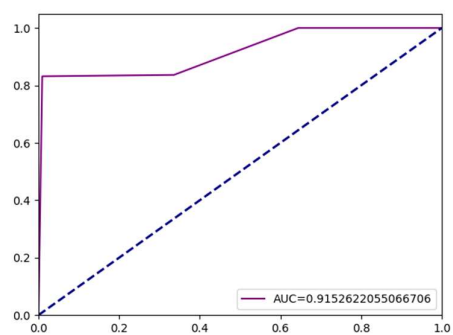
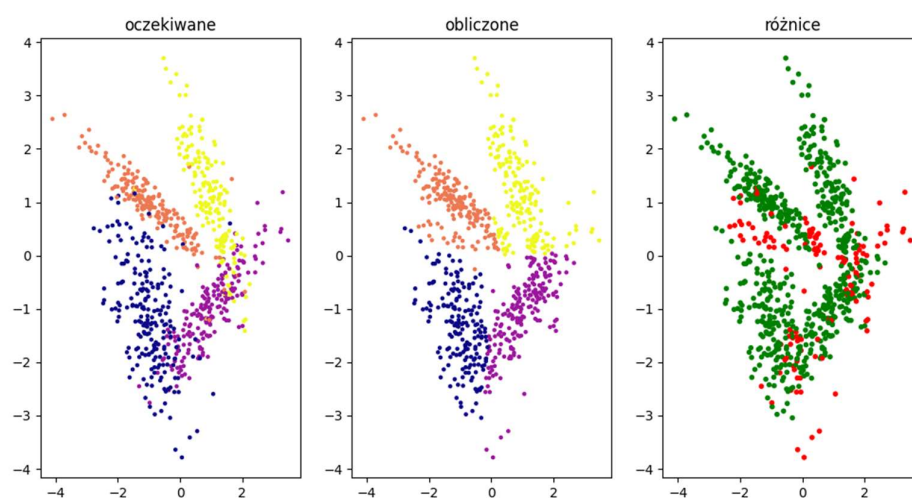
OneVsRestClassifier(), LogisticRegression()



OneVsOneClassifier(), Perceptron()



OneVsRestClassifier(), Perceptron()



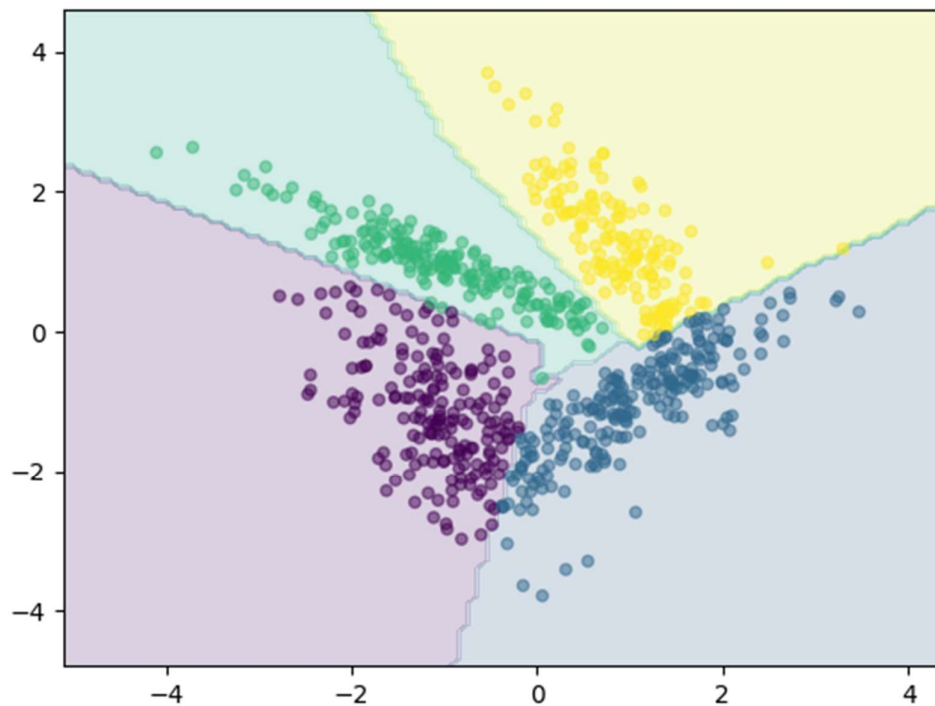
Wykresy wychodzą mi w tego typu przedziałach , więc trudno mi napisać wobec tego jakieś, sensowniejsze wnioski w 2 podpunkcie.

1 OvR osiąga zazwyczaj gorsze wyniki klasyfikacji niż te same parametry dla OvO. Najniższą skutecznością cechuje się OvR dla parametru `Perception()`. Liniowe granice wyznacza SVC z parametrem `kernel` ustawionym na `linear`, a także `Perceptron()` i `LogisticRegression()`.

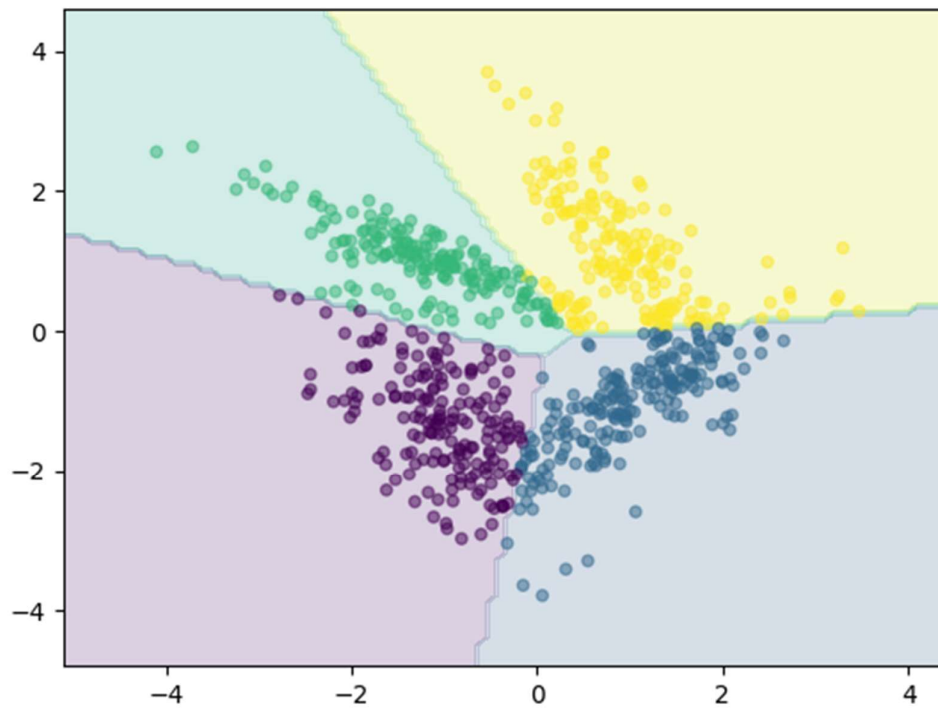
2

Rysunek 3

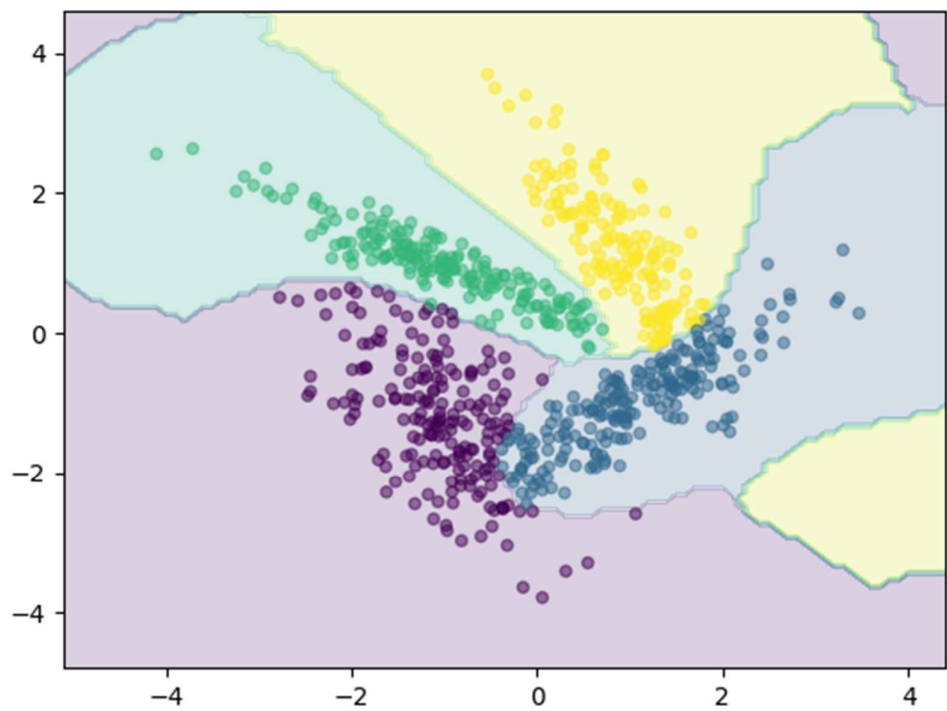
`OneVsOneClassifier()`, `svm.SVC(kernel='linear', probability=True)`



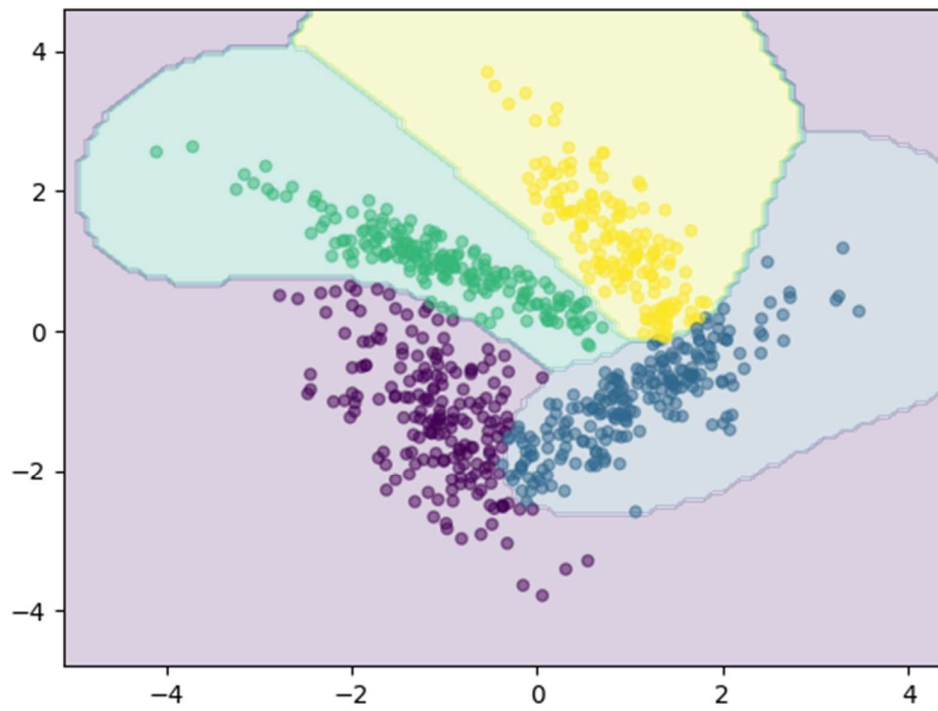
`OneVsRestClassifier()`, `svm.SVC(kernel='linear', probability=True)`



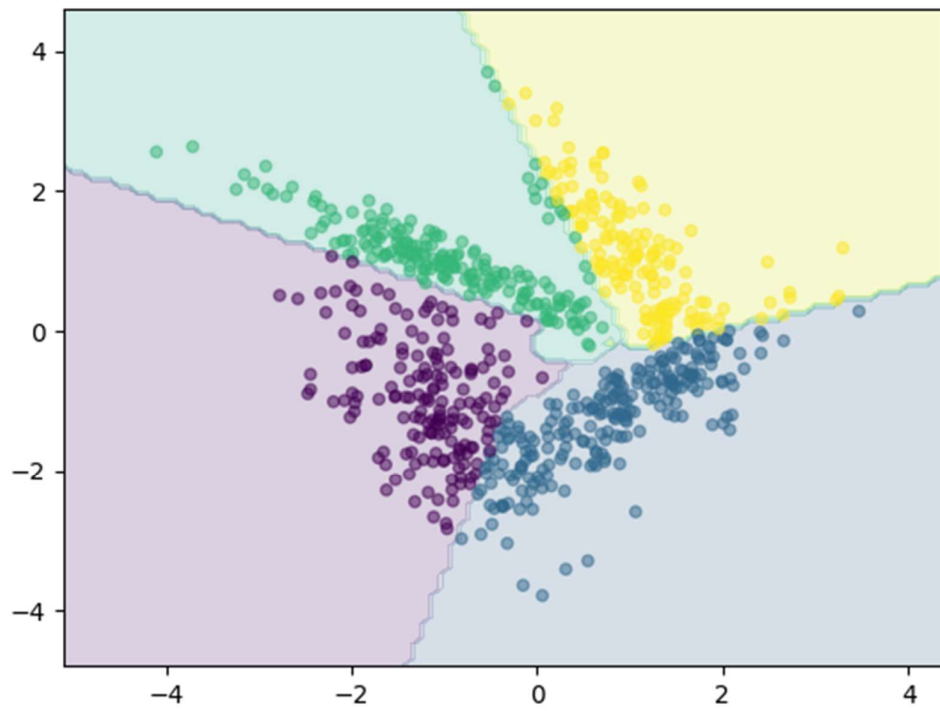
OneVsOneClassifier(), svm.SVC(kernel='rbf', probability=True)



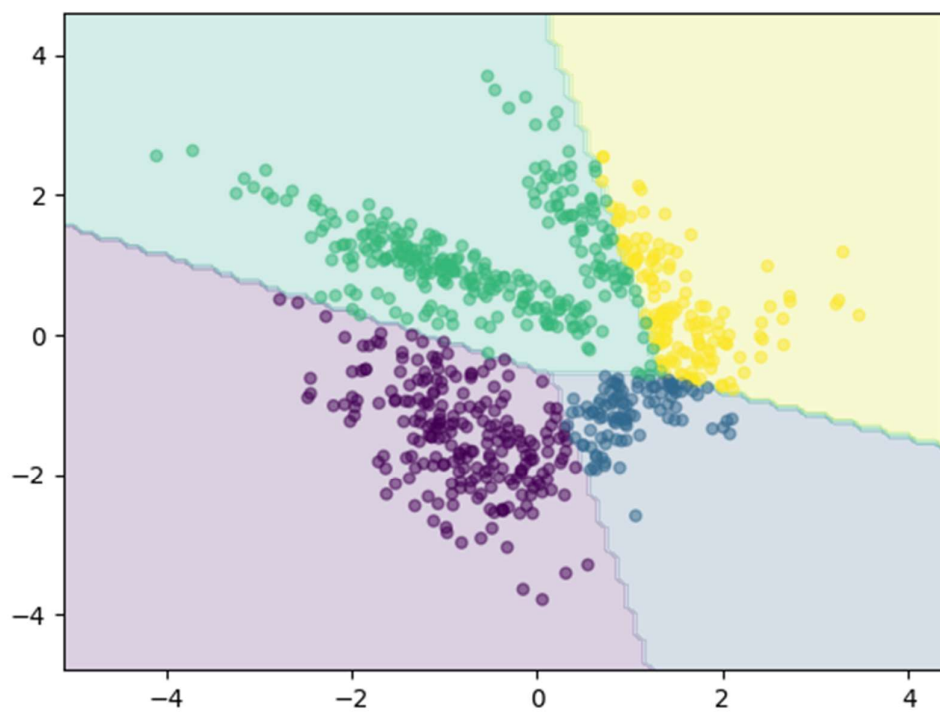
OneVsRestClassifier(), svm.SVC(kernel='rbf', probability=True)



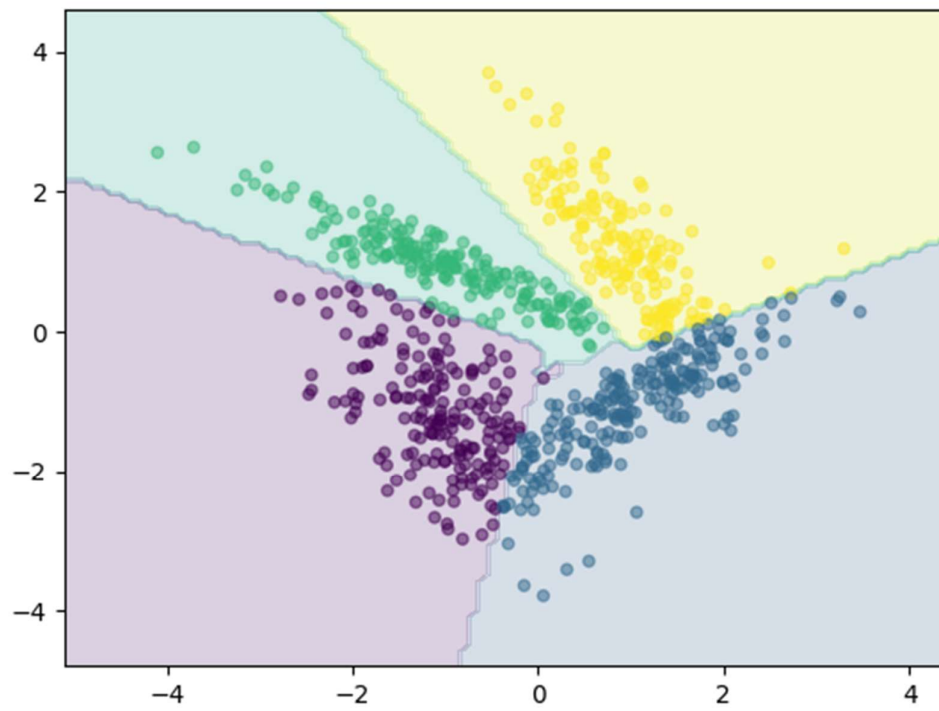
OneVsOneClassifier(), LogisticRegression()



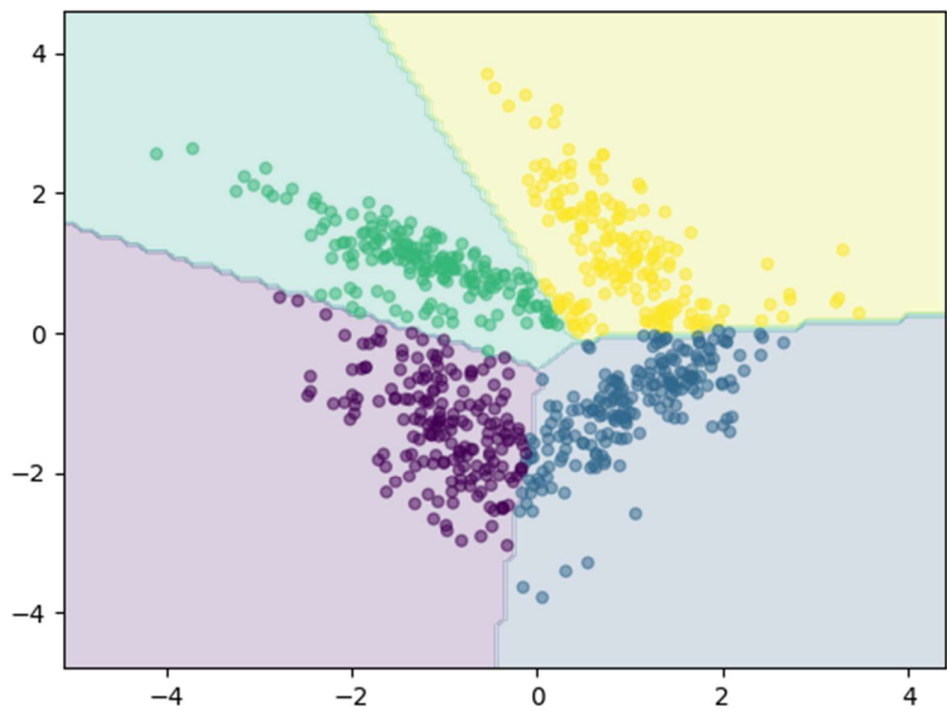
OneVsRestClassifier(), LogisticRegression()



OneVsOneClassifier(), Perceptron()



OneVsRestClassifier(), Perceptron()



Rysunek 4

accuracy_score, recall_score, precision_score, f1_score

