**Temat:** Porównanie klasyfikatorów za pomocą krzywych ROC (dokładnie pól pod wykresem (Area Under Curve - AUC)) w przypadku zadania z wieloma klasami (więcej niż 2)

**Autorzy:**1. Apollo Piotr  
2. Balsam Artur  
3. Marek Dariusz  
4. Śniechota Tomasz  
5. Wierciński Piotr  
6. Zwoliński Jakub

Kroki prowadzące do otrzymania miary M oraz CI przy pomocy klasyfikatora SVM.

1. Pobieranie danych z pliku oraz podzielenie na dane treningowe oraz testowe w stosunku 2/3 do 1/3:

importData <- read.csv(<ścieżka do pliku>), header = TRUE)  
 rowNumber = nrow(importData)  
 train.set = sort(sample(rowNumber, size = round(2/3 \* rowNumber)))  
 test.set = sort(setdiff(seq\_len(rowNumber), train.set))  
 trainData = importData[train.set,]  
 testData = importData[test.set,]  
 rownames(trainData) <- seq(length=nrow(trainData))  
 rownames(testData) <- seq(length=nrow(testData))

2. Tworzenie obiektu "task". Parametry funkcji tworzącej to:

* data - dane treningowe,
* target - nazwa tabeli klasy,

task = makeClassifTask(data = train.set, target = "class")

3. Tworzenie obiektu typu "learner". Parametry funkcji to:

* nazwa klasyfikatora,
* predict.type - typ danych wyjściowych po wykonaniu przewidywań,
* kernel - typ kernela,

lrnSvn = makeLearner("classif.svm", predict.type = "prob", kernel = "sigmoid")

4. Trenowanie oraz przewidywanie klas

model= train(lrnSvn, task)  
 predictions = predict(model, newdata = test.set)

5. Użycie funkcji multiclass.roc i stworzenie obiektu wynikowego. Parametry funkcji:

* wektor klas z danych wejściowych
* wektor klas przewidywanych

predictionsResponse <- as.numeric(getPredictionResponse(predictions))  
 multiclassResult = multiclass.roc(test.set$class, predictionsResponse)

6. Obliczenie miary M przy pomocy

* obiekt otrzymany z funkcji multiclass.roc

aucResult = auc(multiclassResult)

7. Obliczenie miary CI przy pomocy funkcji ci.auc. Parametry funkcji takie same jak w przypadku funkcji multiclass.roc

ci = ci.auc(test.set$class,predictionsResponse)

Ważne informacje zauważone podczas pracy nad projektem:

1. Klasyfikator knn - nie wspiera wyniku przewidywań typu "probe" i trzeba używać "response". Ponadto nie wspiera używania danych nie numerycznych. Podczas pracy nad projektem dane ze zbioru "car" zostały przekonwertowane na dane numeryczne (dane te zostały przesłane również w paczce z danymi).

2. Klasyfikator lda - nie działa na kolumnach o stałej wartości (np. tylko 0). Jeżeli podczas próby losowania dane ułożą się w taki sposób, że jedna z kolumn będzie stała, niestety trzeba powtórzyć losowanie. Podczas samego przygotowywania danych zauważono, że niektóre kolumny danych są stałe dla wszystkich wierszy - przez co musiały zostać usunięte (dane "optdigits")