Przetwarzanie i Analiza Danych

Laboratorium 12: Klasyfikacja wieloklasowa

Cel ćwiczenia

Celem ćwiczenia jest zbadanie możliwości klasyfikacji wieloklasowej za pomocą funkcji dostępnych w pakiecie `scikit-learn`: `OneVsOneClassifier()` i `OneVsRestClassifier()`. Funkcje te opakowują standardowe (binarne) klasyfikatory, dając możliwość pracy na zbiorach zawierających obiekty należące do więcej niż dwóch klas. Strategia OvO sprawdza klasyfikator na wszystkich kombinacjach dwóch wybranych klas (podzbiorach zbioru testowego), natomiast strategia OvR wykorzystuje klasyfikator poprzez "przeciwstawienie" każdej klasy jej dopełnieniu (sumie pozostałych klas ze zbioru).

Zadania

1. Wygenerować przykładowe dane poleceniem `make_classification` z liczbą klas równą cztery, ustalając liczbę atrybutów na dwa. Liczba próbek powinna być większa niż 1500.

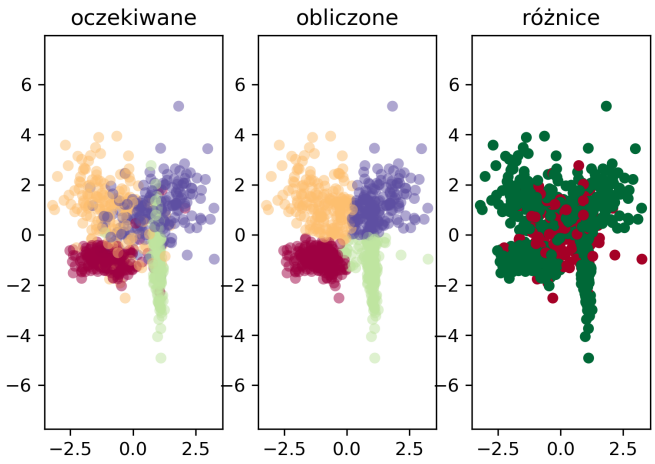
2. Podzielić dane na część uczącą i testującą poleceniem `train_test_split` w proporcjach 50/50.

3. Utworzyć listę 4 par klasyfikatorów w taki sposób aby klasyfikator był "opakowany" w funkcji `OneVsOneClassifier()` i `OneVsRestClassifier()` z przykładowymi parametrami tj:

   - `svm.SVC(kernel='linear', probability=True)`
   - `svm.SVC(kernel='rbf', probability=True)`
   - `LogisticRegression()`
   - `Perceptron()`

4. Dla każdego klasyfikatora z listy i strategii OvO i OvR wykonać kolejne polecenia (powinno to wygenerować 8 niezależnych zbiorów wyników):

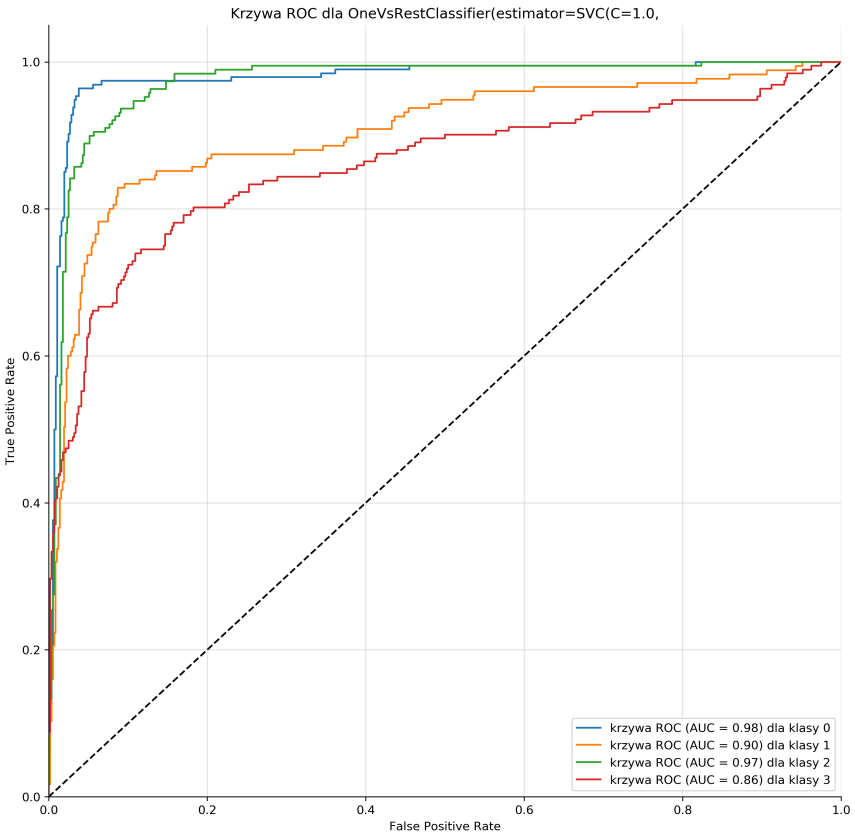
- Wykonać uczenie na zbiorze uczącym (metoda `fit(X_train, y_train)`), a następnie wyznaczyć predykcję na zbiorze testowym: `y_pred = clf.predict(X_test)`
- zwizualizować wyniki klasyfikacji: poprawne i błędne - tak jak na Rysunku 1.



Rysunek 1: Przykładowa wizualizacja zbioru losowych obiektów w 4 klasach - poprawne i błędne

- Wyznaczyć następujące miary jakości klasyfikacji:
  - dokładność (`metrics.accuracy_score`),
  - czułość (`metrics.recall_score`),
  - precyzję (`metrics.precision_score`),
  - F1 (`metrics.f1_score`).
  - pole pod krzywą AUC (`metrics.roc_auc` - uwzględnić, że krzywa ROC obliczana jest zwykle dla klasyfikatora binarnego (tak więc obliczyć wartośc średnia dla każdej z klas))
- Wyznaczyć i narysować krzywą ROC (polecenie `fpr, tpr, thresholds = roc_curve(y_test, y_pred)` - dla każdej z klas indywidualne, zebrać je na jednym rysunku (np. Rysunek 2).
- Narysować powierzchnię dyskryminacyjną (wskazówka: przepuścić przez klasyfikator siatkę punktów wygenerowaną poleceniem `meshgrid`, a następnie użyć poleceń `contour` i `scatter`). Przykład na Rysunku 3.

5. Zwizualizować otrzymane wyniki dotyczące jakości klasyfikacji, jak na Rysunku 4.

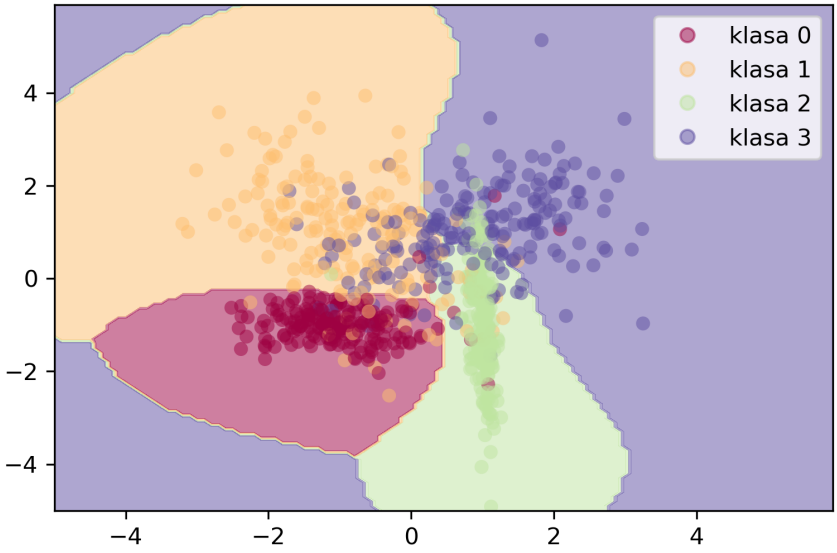


Rysunek 2: Przykładowe porównanie krzywych ROC

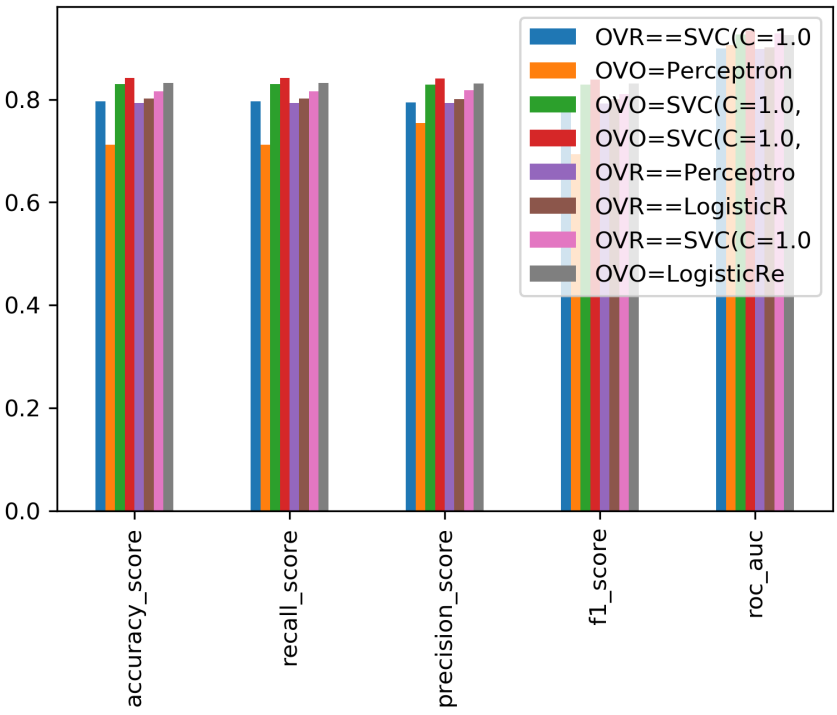
Sprawozdanie - wnioski

Wynik zadań zebrać na 1 stronicowym sprawozdaniu, które powinno zawierać syntetycznie przedstawione wyniki eksperymentów (wykresy, tabele), oraz kończyć się wnioskami dodyczącymi eksperymentu. Zwrócić uwagę an:

- wpływ strategii OvO i OvR na wyniki klasyfikacji (miary jakości oraz krzywe dyskryminacyjne) - które z klasyfikatorów tworzą proste (liniowe) granice decyzyjne a które nie?
- wpływ rozrzutu wygenerowanych próbek należących do konkretnych klas na wskazniki klasyfikacji, głównie AUC -dlaczego dla pewnych klas wskaźnik AUC jest większy niz 0.95 a dla innych mniejszy niż 0.9?



Rysunek 3: Przykładowa wizualizacja krzywych dyskryminacyjncyh i podprzestrzeni związanych z klasami



Rysunek 4: Przykładowe porównanie jakości klasyfikacji