|  | Politechnika Bydgoska im. J.J. Śniadeckich Wydział Telekomunikacji, Informatyki i Elektrotechniki | | |  |
| --- | --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i eksploracja danych | | | |
| **Prowadzący** | dr inż. Michał Kruczkowski | | | |
| **Temat** | Uczenie zespołowe | | | |
| **Studenci** | Adam Szreiber, Cezary Naskręt | | | |
| **Nr ćw.** | 5 | **Data wykonania** | 21.10.2023 | |

# Cel ćwiczenia

Uczenie zespołowe to zaawansowany obszar uczenia maszynowego, w którym wiele modeli jest łączonych w celu poprawy jakości predykcji lub rozwiązania bardziej złożonych problemów.

Laboratorium miało na celu zapoznanie się z podstawowymi metodami uczenia zespołowego. W trakcie laboratorium mieliśmy okazję zapoznać się z różnymi technikami takimi jak Bagging, Random Forest, Boosting, czy Gradient Boosting. Jednym z głównych punktów programu była metoda VotingClassifier, która pozwala na agregację wyników kilku klasyfikatorów w celu uzyskania bardziej stabilnych i dokładnych prognoz.

# Przebieg laboratorium

## Zadanie 1

Zbiór danych: <https://archive.ics.uci.edu/dataset/341/smartphone+based+recognition+of+human+activities+and+postural+transitions>

* Treningowy: X\_train.txt / y\_train.txt
* Testowy: X\_test.txt / y\_test.txt

## Zadanie 2

Wykorzystując zbiór danych z poprzednich laboratoriów wykonaj uczenie zespołowe dla w następującej konfiguracji:

1. Liczba klasyfikatorów: 3
2. Algorytmy klasyfikacji: dowolne
3. Trenowanie: każdy klasyfikator na całym zbiorze treningowym

W ramach uczenia zespołowego wypisz wypadkową ocenę stosując zasadę większości głosów.

Jako rezultat przedstaw wynik kroswalidacji dla 5 podprób oraz wynik testowania na zbiorze testowym.

Jako kryterium oceny klasyfikacji wypisz następujące parametry:

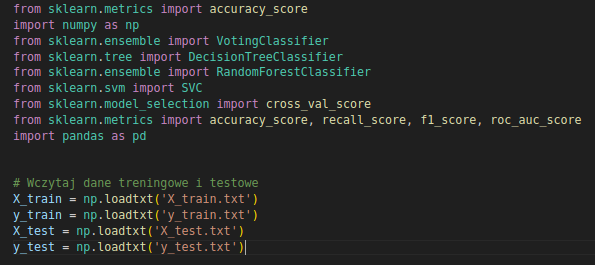
• ACC

• Recall

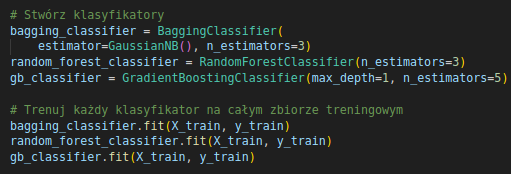
• F1

• AUC

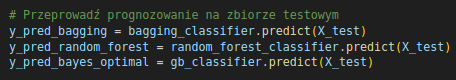
Na początku należało zaimportować wymagane biblioteki i funkcje



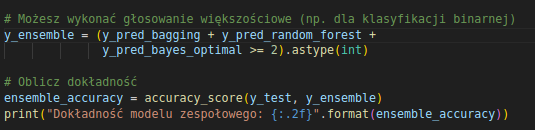
Stworzenie klasyfikatorów i wytrenowanie danymi.



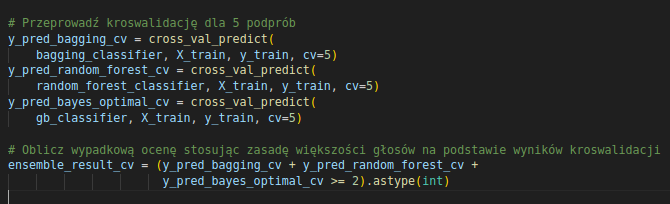
Weryfikacja wytrenowanych modeli na danych testowych



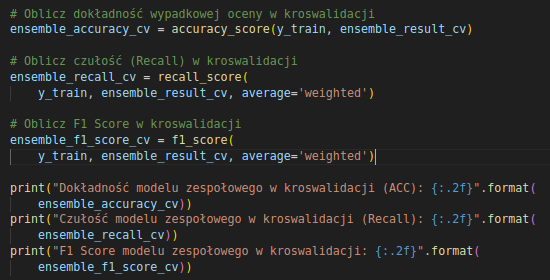
Wykonanie głosowania większościowego i wyświetlenie dokładności



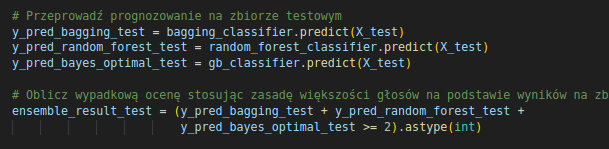
Dodanie kodu kroswalidacji danych i wyliczenie oceny większościowej

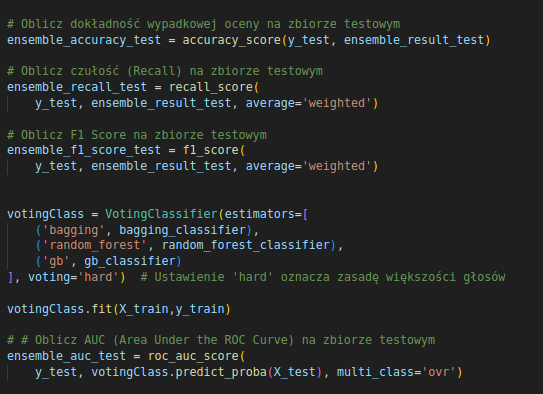


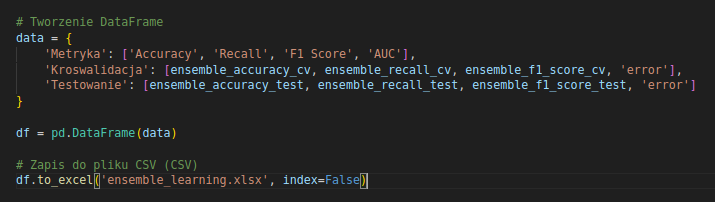
W kolejnym kroku liczymy i wyświetlamy kryteria oceny takie jak: recall score, accuracy score, f1 score.



Należało również przeprowadzić predykcję na zbiorze testowym i ocenić stosując większość głosów co uczyniono poniżej.



Ostatnim krokiem była ocena wyuczonego modelu co przedstawiono poniżej  


Tworzenie obiektu danych w który zostanie zapisany do pliku   


Wyniki wykonywania programu



## Zadanie 3

Wykorzystując zbiór danych z poprzednich laboratoriów wykonaj uczenie zespołowe stosując agregację klasyfikatorów dla w następującej konfiguracji:

4. Liczba klasyfikatorów: 3

5. Algorytmy klasyfikacji: dowolne

6. Trenowanie: na losowo wygenerowanych podzbiorach

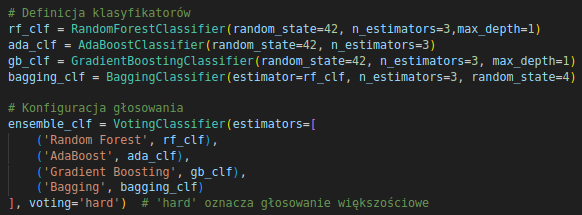
W ramach agregacji klasyfikatorów wypisz wypadkową ocenę stosując zasadę większości głosów.

Jako rezultat przedstaw wynik kroswalidacji dla 5 podprób oraz wynik testowania na zbiorze testowym.

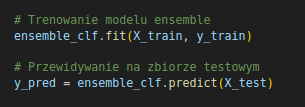
Jako kryterium oceny klasyfikacji wypisz następujące parametry:

* ACC
* Recall
* F1
* AUC

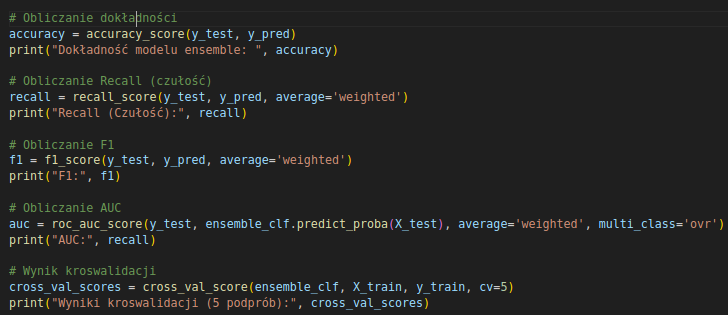
W tym zadaniu pominąłem ładowanie plików ponieważ nic się nie zmieniło względem poprzedniego zadania.  
Poniżej przedstawiono przygotowanie klasyfikatorów i utworzenie klasyfikatora głosowania większościowego zawierającego w sobie następujące klasyfikatory: Random Forest, AdaBoost, Gradient Boosting, Bagging



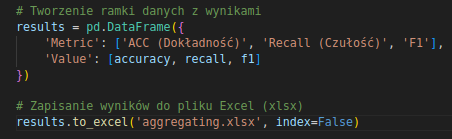
Po utworzeniu klasyfikatorów następnym krokiem było wykonanie trenowania i przewidywania.

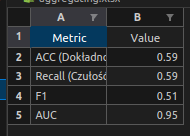


W końcowej części kodu dodano obliczanie oceny klasyfikacji dla metod: accuracy, recall, f1.



Dodanie kodu który wygeneruje plik xlsx i zapisze wyniki.



Poniżej przedstawienie danych zapisanych do pliku excel  


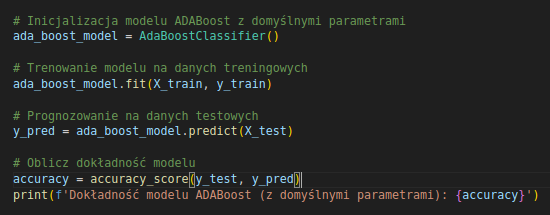
## Zadanie 4

Wykonaj tożsame zadanie klasyfikacji jak w poprzednich ćwiczeniach dla następujących metod uczenia zespołowego przyjmując domyślne parametry:

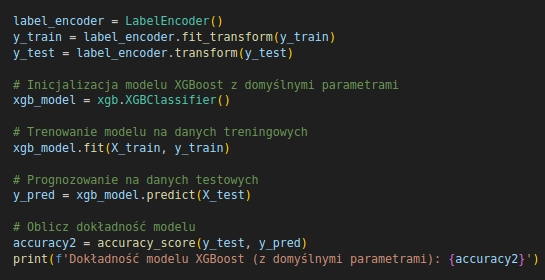
* ADABoost
* XGBoost

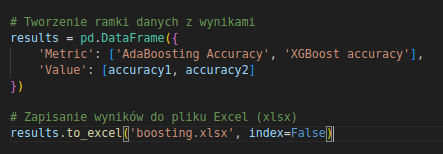
Rezultaty zapisz do pliku o nazwie boosting z rozszerzeniem xls lub xlsx

Tak jak w poprzednim zadaniu zostanie pominięte wczytywanie pliku.  
Poniżej kod realizujący trenowanie, testowanie i sprawdzenie dokładności modelu ADABoost.



Na kolejnym screenie został ukazany kod realizujący inicjalizację, uczenie, prognozowanie, i sprawdzanie dokładności wytrenowania przy pomocy klasyfikatora XGB.



Ostatnim krokiem jest zapisanie wyników do pliku, co wygląda następująco  


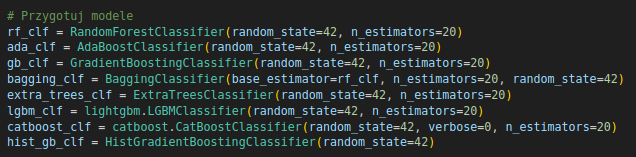
## Zadanie 5

Porównaj wyniki klasyfikacji wszystkich metod uczenia zespołowego w zbiorczej tabeli na podstawie następujących kryteriów:

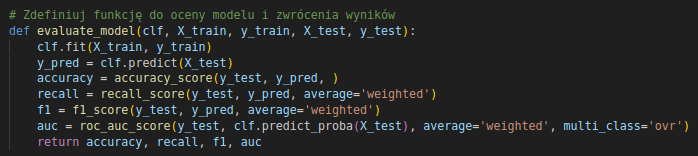
* ACC
* Recall
* F1
* AUC Rezultaty zapisz do pliku o nazwie comparison z rozszerzeniem xls lub xlsx

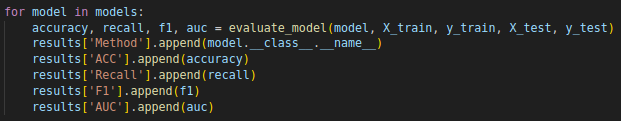
Po raz kolejny pominięto fazę wczytywania danych.

Poniżej kod inicjalizujący modele wszystkie modele uczenia

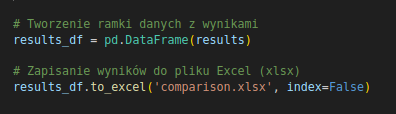


W tym przykładzie postanowiłem zautomatyzować uczenie i testowanie. Cały ten proces został umieszczony w funkcji evaluate\_model

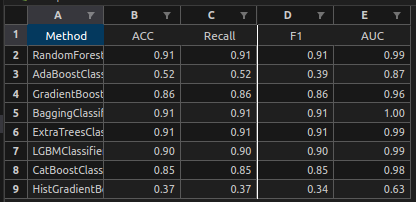


Następnie definiuję tablicę z kalsyfikatorami, w pentli for poruszam się po każdym elemencie tablicy - każdym klasyfikatorze i wywołuję funkcję evaluate\_model z odpowiednimi parametrami. Zwrócone dane zostają zapisane do obiektu results.  


Na podstawie zgromadzonych danych zostaje utworzony DataFrame i zapisany do pliku xlsx.



Widok zapisanego pliku comparison.xlsx.



# Wnioski

Uczenie zespołowe to technika w dziedzinie uczenia maszynowego, która polega na łączeniu wyników wielu modeli, aby uzyskać lepsze wyniki predykcyjne niż pojedynczy model. Jest to jak "gromadzenie mądrości grupy" - poprzez konsultację wielu ekspertów, można dokładniej przewidzieć wyniki.

Głównym celem uczenia zespołowego jest poprawienie jakości predykcji modelu poprzez łączenie wyników wielu modeli bazowych

Poprzez łączenie wyników wielu modeli, uczenie zespołowe może pomóc w zwiększeniu dokładności predykcji. Wielu "ekspertów" pracujących razem może osiągnąć lepsze wyniki niż pojedynczy ekspert.

Redukcja wariancji: Modele bazowe mogą mieć różne błędy i odchylenia od poprawnej prognozy. Poprzez łączenie wyników, model jest bardziej stabilny i mniej podatny na overfitting.

Uczenie zespołowe może poprawić zdolność modelu do generalizacji na nowe dane, co jest kluczowe w zastosowaniach praktycznych

Podczas laboratorium używaliśmy 4 miar oceny klasyfikatora:

Dokładność mierzy ogólną zdolność klasyfikatora do poprawnego przewidywania wszystkich przypadków.

Precyzja mierzy zdolność klasyfikatora do poprawnego wykrywania pozytywnych przypadków.

Recall mierzy zdolność klasyfikatora do wykrywania wszystkich rzeczywistych pozytywnych przypadków.

F1-score jest miarą, która łączy zarówno precyzję, jak i recall w jedną miarę. Jest szczególnie użyteczna w przypadku problemów z niezrównoważonymi danymi.

AUC - (Area Under the Receiver Operating Characteristic Curve) to metryka oceny jakości klasyfikatorów binarnych i wieloklasowych. Pomaga ocenić zdolność klasyfikatora do rozróżniania między klasami poprzez analizę krzywej ROC (Receiver Operating Characteristic). Wyższy wynik ROC AUC oznacza lepszą jakość klasyfikatora, jeśli wynik jest powyżej 0,5, to klasyfikator jest lepszy niż losowy.

VotingClassifier umożliwia łączenie wyników wielu różnych klasyfikatorów w jednym modelu. Szczególnie stosowany w technikach uczenia zespołowego, w którym to wynik jest określany na podstawie większościowego głosowania lub ważonego głosowania.  
Hard Voting (Głosowanie większościowe): uzyskuje największą liczbę głosów (czyli większość) jest wybierany jako ostateczna decyzja modelu zespołowego.

Soft Voting (Głosowanie ważone): Wszystkie modele bazowe przewidują prawdopodobieństwa przynależności do klas, a VotingClassifier sumuje te prawdopodobieństwa ważone wagami i wybiera klasę z najwyższym ważonym prawdopodobieństwem.

RandomForest tworzy wiele drzew decyzyjnych, a następnie łączy wyniki ich głosowania, co prowadzi do modelu o wysokiej odporności na overfitting i zdolności do obsługi dużej ilości danych.

AdaBoost tworzy sekwencję słabych klasyfikatorów i nadaje wagę błędnie sklasyfikowanym przykładom. Każdy nowy klasyfikator jest dostosowywany do błędów poprzednich, co prowadzi do modelu o dużej zdolności do poprawnego klasyfikowania trudnych przypadków.

Gradient Boosting metoda w której model jest tworzony sekwencyjnie poprzez minimalizowanie gradientu funkcji straty. Pozwala to na budowę modelu o dużej zdolności do generalizacji i dobrze radzącego sobie z problemami klasyfikacji i regresji.

Bagging (Bootstrap Aggregating) to technika, w której wiele niezależnych klasyfikatorów jest trenowanych na losowo wygenerowanych podzbiorach danych treningowych.

Extra Trees to wariant algorytmu Random Forest, w którym drzewa decyzyjne są tworzone na podstawie losowych podzbiorów cech i próbek. Jest bardziej losowy niż Random Forest, co pomaga w zmniejszeniu wariancji modelu.

LightGBM wydajny algorytm uczenia zespołowego oparty na drzewach decyzyjnych. Wykorzystuje techniki przetwarzania gradientowego i histogramowego, co pozwala na szybsze trenowanie i uzyskanie modeli o wysokiej dokładności.

CatBoost został zaprojektowany specjalnie do obsługi danych kategorycznych.

HistGradientBoosting opiera się na drzewach decyzyjnych i wykorzystuje histogramy do przyspieszenia trenowania.