# Wstęp do uczenia maszynowego - projekt

Oliwia Trzcińska

## 1 Wstęp

Celem projektu jest stworzenie modelu klasyfikacji o jak największej mocy predykcyjnej. Będziemy zajmować się danymi ze sztucznie wygenerowanego zbioru `artifical`. Dokonamy klasyfikacji do dwóch klas. Dokładność modelu będziemy oceniać na podstawie miary zrównoważonej dokładności (*balanced accuracy*).

## 2 Dane

Zbiór danych `artifical` zawiera 30 zmiennych. Zbiór treningowy ma 2000 obserwacji, a zbiór testowy - 600. Znamy tylko etykiety obserwacji ze zbioru treningowego.
Wszystkie zmienne są numeryczne. W żadnym ze zbiorów nie ma braków danych. Histogramy większości zmiennych przypominają rozkład normalny. Nie występują też widoczne obserwacje odstające.
Ze względu na brak znajomości etykiet obserwacji z wcześniej wspomnianego zbioru testowego, do dalszej pracy podzielimy zbiór treningowy na część treningową i testową w proporcji 9:1. W ten sposób zostały stworzone zbiory `X_train_train`, `X_train_test` oraz wektory etykiet `y_train_train` i `y_train_test`.
Przed rozpoczęciem procedury wyboru najlepszego typu modelu klasyfikacji została przeprowadzona analiza głównych składowych - PCA. Celem jej zastosowania była redukcja wymiaru danych. Okazało się jednak, że nie polepsza ona wyników, na naszych danych działa wręcz przeciwnie. Z tego powodu pomysł z zastosowaniem PCA został odrzucony. Na tym etapie tworzymy modele nie usuwając żadnych (być może nieistotnych) zmiennych.

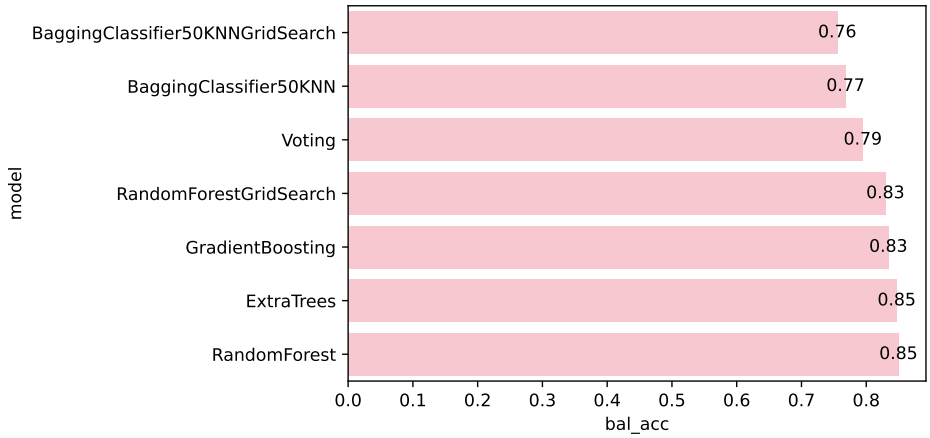
## 3 Wybór modelu

Najpierw przeprowadzimy eksperyment. Jego celem będzie wybór typu modelu, który osiągnie najwyższą wartość zrównoważonej dokładności w zadaniu klasyfikacji. Bierzemy pod uwagę 7 modeli:

- komitet składający się z 50 modeli $k$ najbliższych sąsiadów,
- komitet składający się z 50 modeli $k$ najbliższych sąsiadów; optymalizujemy wartość parametru `k`,
- las losowy,
- las losowy; optymalizujemy liczbę użytych drzew decyzyjnych,
- model `ExtraTrees`,
- model `GradientBoostingClassifier`,
- model `VotingClassifier`, korzystający z modeli regresji logistycznej, lasu losowego i $k$ najbliższych sąsiadów.

Przy tworzeniu tych modeli optymalizujemy tylko wymienione hiperparametry. Używamy do tego celu funkcji `GridSearchCV` z trzykrotną walidacją. Maksymalizujemy wartość *balanced accuracy*. Modele uczymy na całym zbiorze danych `X_train_train`. Ich jakość oceniamy mierząc *balanced accuracy* na części treningowej i testowej zbioru danych (`X_train_train` oraz `X_train_test`).

Miary jakości poszczególnych modeli na zbiorze testowym `X_train_test` zostały przedstawione na rysunku 1.




Rysunek 1: Zależność wartości zrównoważonej dokładności od typu modelu

Najlepsze wyniki na części testowej zbioru danych uzyskały modele `RandomForest` oraz `ExtraTrees`. Skupimy się więc teraz na ulepszaniu tych dwóch modeli.

# 4 Model lasu losowego

Będziemy teraz dążyć do wyeliminowania ze zbioru danych zmiennych nieistotnych oraz zbudowania jak najlepszego modelu lasu losowego na niepełnych danych.
Najpierw tworzymy model lasu losowego bez ustawiania innych niż domyślne wartości parametrów. Dopasowujemy go do całych danych `X_train_train`. Następnie korzystamy z funkcji `SequentialFeature Selector`. Stosujemy trzykrotną kroswalidację i maksymalizujemy wartość zrównoważonej dokładności. Istotnych okazało się dzięki temu działaniu 15 zmiennych.
Teraz ponownie tworzymy taki sam model lasu losowego, ale dopasowujemy go do danych z wyselekcjonowanymi wcześniej zmiennymi. Dzięki temu poprawiliśmy *balanced accuracy* dla danych z części testowej do wartości 0,887.
Sprawdzimy teraz, jak poprawi się jakość modelu lasu losowego, jeśli wybierzemy istotne zmienne przy pomocy funkcji `SelectFromModel`. Przeprowadzamy podobną procedurę jak poprzednio. Dzięki niej okazuje się, że aż 22 zmienne są nieważne dla modelu. Tworzymy więc nowy las losowy dopasowany do danych treningowych z wyselekcjonowanymi tą metodą zmiennymi. Poprawiamy tym samym wartość *balanced accuracy* dla danych testowych do wartości 0,902. Jest to jak dotąd najwyższy wynik.

# 5 Model `ExtraTrees`

Ponownie dążymy do wyeliminowania ze zbioru danych nieistotnych zmiennych. Tym razem zamiast lasu losowego wykorzystujemy model `ExtraTrees`.
Przeprowadzamy taką samą procedurę jak wcześniej. Korzystając z funkcji `SequentialFeatureSelector` eliminujemy 15 zmiennych. Dzięki temu *balanced accuracy* osiąga na części testowej zbioru wartość 0,865.
Skorzystanie z funkcji `SelectFromModel` prowadzi z kolei do uznania 22 zmiennych za nieistotne oraz wartości *balanced accuracy* równej 0,893.

# 6 Podsumowanie

Zarówno podczas procedury tworzenia modelu, jak i selekcji zmiennych podejmowałam próby optymalizacji różnych hiperparametrów. Za każdym razem wynik osiągany na zbiorze testowym nie był

znacząco lepszy niż bez optymalizacji parametrów. Dlatego stworzyłam las losowy z domyślnymi wartościami parametrów, następnie przy pomocy funkcji `SelectFromModel` wybrałam ze zbioru danych istotne zmienne i ponownie dopasowałam do nich model lasu losowego. Dzięki temu ostateczna wartość *balanced accuracy* na zbiorze `X_train_train` wynosi 1, a na zbiorze `X_train_test`: 0,902. Na 5% „prawdziwego" zbioru testowego wynik to 0,9.
Na koniec stworzyłam taki sam model, jednak dopasowałam go do wszystkich danych ze zbioru treningowego. Osiągnął on zrównoważoną dokładność na 5% zbioru testowego równą 0,9.