

Stworzony pakiet obsługuje działania klasyfikacyjne, zrobione za pomocą drzew i lasów decyzyjnych. Są dwa typy klasyfikatorów: zwykły oraz z wykorzystaniem regresji. Odpowiednio są obsługiwane klasy *RandomForestClassifier* oraz *RandomForestRegressor*. Każdy z konstruktorów przyjmuje jeden parameter – *n\_features*, który ma być liczbą naturalną.

Metody dla *RandomForestClassifier*:

- *fit(X, y)* - uczy klasyfikator na zbiorze treningowym *X* (*y* jest wektorem, który dla każdego wiersza *X* zawiera klasę, do której należy ten przykład)
- *predict(X)* - przewiduje najbardziej prawdopodobne klasy przykładów w *X*; wraca wektor długości *m*
- *predict\_proba(X)* - zwraca wektor (rozmiaru *m x 1*) prawdopodobieństw przynależności przykładów z *X* do pierwszej klasy (za pierwszą klasę rozumiemy tutaj klasę występującą w zbiorze treningowym jako pierwsza)

Metody dla *RandomForestRegression*:

- *fit(X, y)* - uczy klasyfikator na zbiorze treningowym *X* (*y* jest wektorem, który dla każdego wiersza *X* zawiera wartość zmiennej zależnej)
- *predict(X)* - wynik regresji dla przykładów w *X*; zwraca wektor liczb rzeczywistych długości *m*

Dany klasyfikator jest w stanie pracować jak z parametrami liczbowymi, tak i z parametrami mieszanymi (np. tekstem, datami). W przypadku danych liczbowych porównywanie idzie według warunku mniej lub więcej. Dla danych mieszanych porównywanie idzie według warunku równe lub nie równe.

Dla zwykłego klasyfikatora przez to, że decyzja jest arbitralna pojawia się funkcja *predict\_proba*, która pokazuje prawdopodobieństwo przynależności przykładów do pierwszej klasy. Dana funkcja analizuje las losowy (minimalnie 11 drzew) i wylicza prawdopodobieństwo dla każdego drzewa i wiersza; zwraca listę prawdopodobieństw. Funkcja *predict* zwraca wektor decyzji dla danego lasu losowego, przechodząc po całym lesie i spisując decyzje. Jeżeli więcej połowy decyzji są pozytywne – końcowa decyzja jest pozytywna.

Dane funkcje korzystają z funkcji pomocniczych: *buildTree* (zwraca drzewo zbudowane na podstawie losowo zbudowanej macierzy, żeby uniknąć pobierania jednotypowych decyzji), *build\_random\_forest* (buduje las losowy, korzystając z *buildTree*, dopóki nie ustabilizuje się OOB), *check\_decision\_proportion* (sprawdza czy proporcja klas jest podobna do zbioru testowego), *find\_ooberr* (zwraca wartość OOB), *fit* (funkcja, ucząca klasyfikator).

Dla klasyfikatora z regresją *predict* zwraca najbardziej prawdopodobną decyzję. Wynikiem jest wektor długości początkowej macierzy. Dana klasa korzysta w całości z metod opracowanych w klasie *RandomForestClassifier* i korzysta z jej funkcji pomocniczych.

Oprócz tych klas jest klasa *Test*, która zawiera następujące funkcje:

*k\_mers*: tworzy wszystkie możliwe k-mery. Defaultowo *k=4*.

build\_test\_string\_set: pobiera dane z plików enhancers\_heart.fa i random.fa oraz zlicza wszystkie 4-mery i dla danych z pliku z enhancerami przypisuje wartość True, a dla losowych – False.