METODY EWOLUCYJNE I UCZENIE SIĘ MASZYN - SPECYFIKACJA ZADANIA

AGNIESZKA CZAPLICKA, BARTOSZ SOWUL

Temat: Na kilku zadaniach klasyfikacji z repozytorium UCI porównać metody xgboost i catboost.

1 OPIS ZAGADNIENIA I PRZYJĘTE ZAŁOŻENIA

Celem projektu jest porównanie bibliotek *catboost* i *xgboost* w zadaniach klasyfikacji. Projekt zostanie zrealizowany w języku *Python 3* z wykorzystaniem aplikacji webowej *Jupyter*. Do optymalizacji hyperparametrów algorytmów zostanie użyta metoda *RandomizedSearchCV* z biblioteki *sklearn* lub biblioteka *hyperopt* (optymalizacja bayesowska).

2 OPIS ALGORYTMÓW

2.1 XGBOOST

Biblioteka *XGBoost* (*E**X**treme **G**radient **Boost**ing*) implementuje algorytm drzew decyzyjnych ze wzmocnieniem gradientowym (*gradient boosting decision tree algorithm*). *Boosting* to technika typu *ensemble*, bazująca na założeniu, że model zbudowany na podstawie wielu prostszych modeli, pozwoli uzyskać lepsze wyniki od każdego z nich. W tym przypadku nowy model wie, jakie błędy popełniały modele wcześniej istniejące. Każde kolejno budowane drzewo stara się podejmować decyzje w taki sposób, aby zminimalizować błędy drzew poprzednich. W związku z tym, że funkcja opisująca dane wykorzystane do nauki nie jest znana, do minimalizacji tego błędu wykorzystywana jest metoda gradientu prostego.

XGBoost był dużo szybszy od pozostałych ówcześnie używanych algorytmów typu *gradient boosted trees*, między innymi dlatego, że przechowuje on informację o niezerowych próbkach (w przypadku cech kategorycznych, w większości przypadków dane to głównie 0) i przy podejmowaniu decyzji o podziale tylko one są brane pod uwagę. Ponadto *XGBoost* sortuje próbki w kolumnach niezależnie od siebie, co pozwoliło na łatwe zrównoleglenie tego procesu. Zamiast dokładnego sortowania wszystkich próbek, algorytm dzieli je na "koszyki", więc nie musi przechowywać całego zbioru danych w pamięci. Przy wystarczająco dużej liczbie takich "koszyków", wyniki algorytmu są porównywalne do tych, zwracanych przez dokładne sortowanie, ale uzyskiwane są zdecydowanie szybciej.

Algorytm ten działa tylko na cechach ilościowych.

2.2 CATBOOST

W przeciwieństwie do algorytmu *xgboost*, *catboost* (***cat**egorical **boost**ing*) działa również z cechami kategorycznymi. Przed każdym wybraniem podziału w drzewie, cechy nieliczbowe

transformowane są na cechy ilościowe. Odbywa się to według następującego algorytmu:

- obiekty wejściowe są kilkukrotnie permutowane,
- cechy jakościowe nominalne zamieniane są na cechy numeryczne zgodnie ze wzorem $avg\_target = \frac{countInClass+prior}{totalCount+1}$, gdzie
  - *countInClass* oznacza, ile razy etykieta wynosiła 1, tzn. należy do klasy,
  - *prior* jest ustalane na podstawie parametrów startowych,
  - *totalCount* to ilość zliczeń obiektów (do obiektu przetwarzanego), które są z tej samej klasy.

  Oznacza to, że wartości numeryczne dla każdego obiektu liczone są tylko na podstawie wcześniejszych obiektów.

3 Optymalizowane parametry

3.1 XGBOOST

- learning_rate (eta) - współczynnik uczenia
- max_depth - maksymalna głębokość drzewa
- min_child_weight - minimalna suma wag przykładów, przekazywanych w wyniku podziału do nowego węzła, wymagana do przeprowadzenia dalszego podziału
- gamma - minimalna zmiana funkcji loss wymagana do przeprowadzenia dalszego podziału
- subsample - procent danych próbkowany przed rozbudową drzew
- colsample_bytree - procent wybranych kolumn użytych do konstrukcji drzewa
- reg_alpha - współczynnik regularyzacji L1
- reg_lambda - współczynnik regularyzacji L2
- n_estimators - liczba drzew

3.2 CATBOOST

- learning_rate - odpowiada za wielkość kroku, im mniejsza tym więcej operacji potrzeba do zbudowania modelu
- depth - głębokość drzewa
- l2_leaf_reg - współczynnik regularyzacji L2, pozwala zmniejszyć zjawisko overfittingu

- rsm - random subspace method, określa jaki procent cech będzie użyty przy wyborze podziału
- random_strength - mnożnik wariancji zmiennej losowej dodawanej do wyniku każdego splitu
- iterations - maksymalna liczba drzew jaka może być użyta do rozwiązania problemu

4 Wykorzystane zbiory danych

Do testów algorytmów zostaną wykorzystane zbiory danych z repozytorium *UC Irvine Machine Learning Repository*.

4.1 Adult

Zbiór składa się z 48842 próbek o 14 cechach, zarówno ilościowych o wartościach ciągłych (6), jak i jakościowych (8). Zadanie polega na predykcji, czy zarobki danej osoby przekraczają kwotę 50 tysięcy USD rocznie. Stosunek przykładów klasy pozytywnej(>50K) do negatywnej(<=50K) wynosi 1:3.

4.2 Annealing

Zadanie polega na określeniu, przy użyciu której z 6 metod wyżarzania metali, uzyskany został metal o zadanych właściwościach. Zbiór składa się z 798 próbek o 38 cechach, w tym: 3 całkowitoliczbowe, 6 zmiennoprzecinkowych i 29 opisowych. Rozkład klas prezentuje się następująco:

Nazwa klasy	Liczba przykładów
1	8
2	88
3	608
4	0
5	60
U	34

4.3 Breast Cancer Wisconsin

Zbiór danych został obliczony na podstawie obrazu materiału uzyskanego z biopsji aspiracyjnej cienkoigłowej.

Zbiór składa się z 699 próbek o 9 cechach, przyjmujących wartości całkowite z zakresu 1-10. W tym zadaniu algorytm klasyfikuje typ guza (łagodny, złośliwy). Przykłady reprezentujące klasę 'łagodny' stanowią 65,5% całego zbioru, a klasę 'złośliwy' - 34,5%.

4.4 CARDIOTOCOGRAPHY

Zbiór cech uzyskany został z przetworzonych kardiotokogramów. Kardiotokografia (KTG) to monitorowanie czynności serca płodu z jednoczesnym zapisem czynności skurczowej macicy. Badanie jest wykonywane pod koniec i w trakcie porodu, pozwala wykryć sytuacje zagrożenia życia płodu.

Zbiór składa się z 2126 próbek o 21 cechach o charakterze ilościowych, z czego 8 przyjmuje wartości ciągłe, a pozostałe dyskretne. W tym zadaniu algorytm klasyfikuje stan płodu (normalny - 1, podejrzany - 2, patologiczny - 3). Rozkład przykładów reprezentujących poszczególne klasy przedstawiono w poniższej tabeli:

Nazwa klasy	Liczba przykładów
1	1655
2	295
3	176

5 PRZEWIDYWANE WYNIKI PRACY

Przy użyciu algorytmów *xgboost* oraz *catboost* zostaną stworzone modele predykcji dla poszczególnych zagadnień, a ich działanie zweryfikowane na nowych danych. W związku z tym, że nie w każdym zadaniu dostępny jest zbiór testowy, eksperymenty przeprowadzimy w oparciu o metodę K-krotnej walidacji krzyżowej. Wybrane zbiory danych zostaną poddane wstępnemu przetworzeniu, a następnie podzielone na K podzbiorów, z których każdy będzie kolejno stanowił zbiór testowy. Na pozostałych K-1 podzbiorach wytrenowany zostanie model, którego zdolności klasyfikacyjne zostaną ocenione na danych niewykorzystywanych do nauki.

Uzyskane wyniki zostaną wykorzystane do porównania algorytmów pod względem poprawności klasyfikacji. Kryterium oceny skuteczności algorytmów będą stanowiły dokładność klasyfikacji (*accuracy*), parametr *F1 score* i wartość metryki *logloss* na danych, stanowiących w danym momencie zbiór testowy. Drugim ważnym rozpatrywanym przez nas aspektem będzie szybkość uczenia, a więc długość trwania tego procesu do momentu osiągnięcia warunków zatrzymania.