Optymalizacja hiperparametrów xgboost

Dokumentacja wstępna

Przemysław Stawczyk, Piotr Zmyślony

12 kwietnia 2020

Spis treści

1 Treść zadania

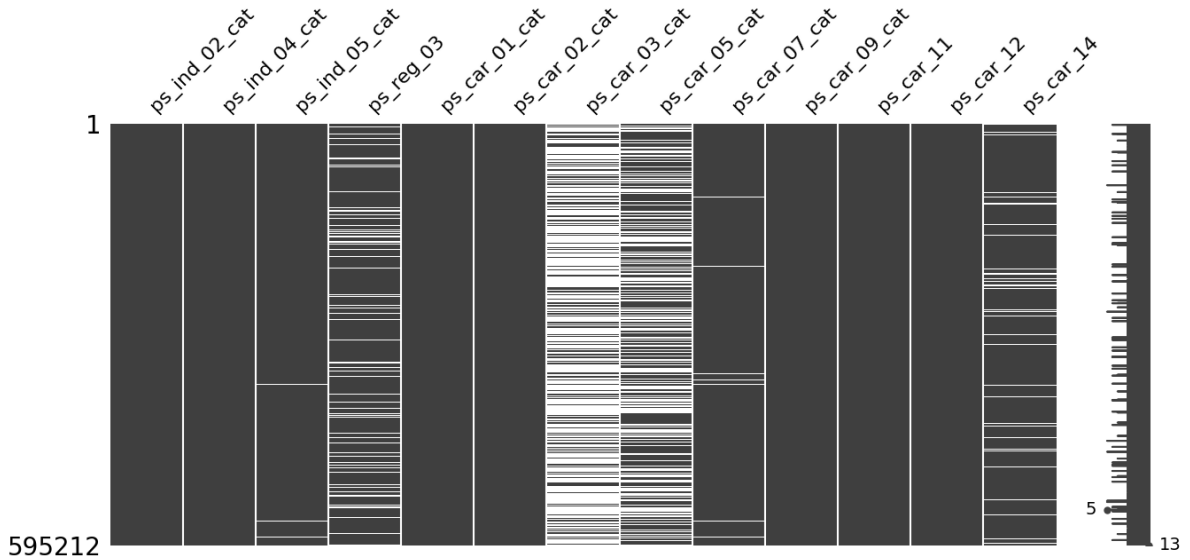
Naszym zadaniem jest przetestowanie różnych algorytmów heurystycznych/populacyjnych w kontekście problemu strojenia hiperparametrów algorytmu xgboost. Problem wyboru hiperparametrów wynika z ich bardzo dużej ilości, co często rozwiązyne jest poprzez manualny dobór parametrów klasyfikatora.

Projekt zostanie zrealizowany w języku Python 3+.

2 Dane testowe

Jako dane na których będziemy trenować i testować klasyfikatory przyjęliśmy proponowany zestaw danych https://www.kaggle.com/c/porto-seguro-safe-driver-prediction. Zawiera on 57 atrybutów opisujących klientów firmy ubezpieczeniowej i jeden atrybut binarny sygnalizujący, czy w ciągu roku od zawarcia umowy, klient skorzystał z ubezpieczenia.

Rysunek 1: Brakujące atrybuty



2.1 Analiza danych

Po wstępnej analizie danych odkryliśmy, że w zbiorze danych posiadamy około 79% niekompletnych wierszy. Rysunek 1 przedstawia pokrycie niekompletnych atrybutów - jest ich jedynie 13, z czego większość jest wybrakowana w bardzo niewielkim stopniu.

Największym winowajcą jest atrybut binarny *ps_car_03_cat*, którego brakuje aż w 70% wierszy, oraz atrybut *ps_car_05_cat* (brakuje go w 44% przypadków).

2.2 Uzupełnienie brakujących danych

W związku z powyższym, planujemy uzupełnić brakujące atrybuty na bazie kompletnych wierszy danych. Do tego zastosujemy bibliotekę pythonową *impyute*, ale nie będziemy analizować, jaka jest zależność między konkretnymi metodami interpolacji wybrakowanych atrybutów a hiperparametrami trenowanego klasyfikatora - ręcznie wybierzemy tą, która daje najlepsze (i najszybsze) rezultaty.

2.3 Alternatywne dane

Końcową wersję naszego algorytmu heurystycznego planujemy przetestować przy użyciu dodatkowego zbioru danych dot. przewidywania bankructwa polskich firm, który analizowaliśmy w innym projekcie.

3 Propozycja rozwiązania

//////////////////TODO: chyba faktycznie weźmy algo wspinaczkowy z tabu i bez plus może jeszcze jakiś jeden na pałę, dla porównania - IMO genetyczny jakiś

4 Funkcja celu

//////////////////TODO: skoro mamy firme ubezpieczeniową, to Expected cost jest chyba najlepszy, tu jest ladnie opisany: https://www.svds.com/the-basics-of-classifier-evaluation-part-1/

Expected cost = p(p) × [p(true positive) × benefit(true positive) + p(false negative) × cost(false negative)] + p(n) × [p(true negative) × benefit(true negative) + p(false positive) × cost(false positive)]

Pewnie przydaloby sie znalezc jakies typowy zysk na 1 rok z ubezpieczenia (zysk dla firmy) i sredni koszt wyplaty ubezpieczenia jednemu klientowi.

5 Sposób mierzenia jakości rozwiązania

//////////////////TODO: czy to nie dokladnie to samo co po prostu funkcja celu? czy moze chodzi o to zeby opisać jak sprawdzamy to (k-krotne walidacje itp.). Chociaz deja napisal to *"sposobu mierzenia jakości rozwiązania (podsumowania wyników)."* i nie wiem co ma podsumowanie wynikow do tego sposobu mierzenia jakosci rozwiazania, moze ty lepiej zrozumiesz.