Cross validation w nauczaniu nadzorowanym

Modelowanie to tworzenie środowiska wirtualnego, które ma w jakimś stopniu opisywać rzeczywistość. Opis ten jest potrzebny, aby zrozumieć przebieg jakiegoś procesu. Czasami tworzymy model, aby zaobserwować jakąś prawidłowość, wychwycić pewne anomalie lub zjawiska niepożądane. Staramy się zrozumieć zjawiska, aby umieć przewidywać ich przebieg w przyszłości.

W tworzeniu modeli dominują dwa podejścia:

- nauki modelu z nadzorem
- nauki modelu bez nadzoru.

W publikacji będziemy mówili o nauce z nadzorem, która w dużym uproszczeniu polega na trenowaniu modelu na zbiorze treningowym i sprawdzaniu efektów na zbiorze testowym. Pierwszym krokiem w nauce nadzorowanej jest podział zbioru danych na mały testowy i duży treningowy.

Jak wiadomo typowy zbiór danych składa się ze zmiennych opisujących (zwanych również zmiennymi niezależnymi) oraz z danych wynikowych, czyli wyników opisujących zjawisko. Np. wynikiem może być poziom zużytego paliwa w samochodzie, zaś zmiennymi opisującymi ten wynik mogą być takie informacje jak długość przejechanej trasy, czynniki atmosferyczne, obciążenie lub prędkość z jaką poruszał się pojazd.

Model zbiera dane opisujące, analizuje ich wielkości, zestawia je z wartościami wynikowymi, w ten sposób uczy się procesu, i na koniec jest zdolny do przewidywania teoretycznej wysokości zużycia paliwa. Aby model mógł wygenerować wyniki wystarczy podstawić do niego zmienne niezależne.

Jak sprawdzić czy model działał poprawnie?

Jeżeli mamy tylko dane treningowe to jedynym sposobem na sprawdzenie jakości tworzonych prognoz jest użycie modelu w praktyce. A co, jeżeli mamy 20 różnych modeli opisujących ten sam proces? W jaki sposób dowiedzieć się, który model jest najlepszy?

Aby oceniać jakość modeli zbiór danych dzieli się losowo na treningowy i testowy[1]. Model trenowany jest na danych treningowych. Po zakończeniu treningu do gotowego modelu podstawia się testowe zmienne niezależne. Wytrenowany model dostaje nieużywane wcześniej w procesie nauki zmienne opisujące i na ich podstawie generuje teoretyczne wyniki. Zbiór testowy zawiera również wyniki empiryczne i możemy porównać je z danymi, które wygenerował model. W ten sposób możemy ocenić jakość modelowania. Tak w dużym uproszczeniu działa metoda nauki z nadzorem.

Przeuczenie modelu

Istnieje wiele różnych algorytmów modelowania procesów. Niektóre z nich mają niski poziom dopasowania do wyników empirycznych, ale nie odbiegają od nich istotnie; inne algorytmy mają z kolei tendencje do bardzo wysokiego poziomu dopasowania do zmiennych empirycznych. Przy modelach mających tendencje do osiągania wysokiego poziomu dopasowania wyników teoretycznych do empirycznych najczęściej występuje problem przeuczenia.

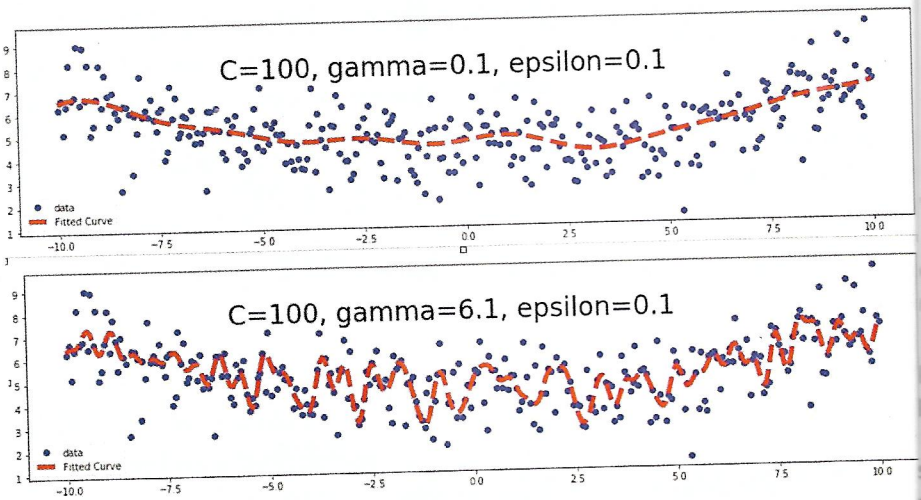
Klasycznym przykładem przeuczenia jest sytuacja, gdy model doskonale odzwierciedla zjawisko w procesie nauki, osiągając bardzo wysoki poziom dopasowania wyników bazując na danych treningowych. Jednocześnie ten sam model, operując na zmiennych testowych, wykazuje niski poziom dopasowania wartości teoretycznych do wyników empirycznych. Model przeuczony nie nadaje się do wykorzystania w praktyce. Jest to dość powszechny problem w procesie modelowania.

Jak likwidować zjawisko przeuczenia

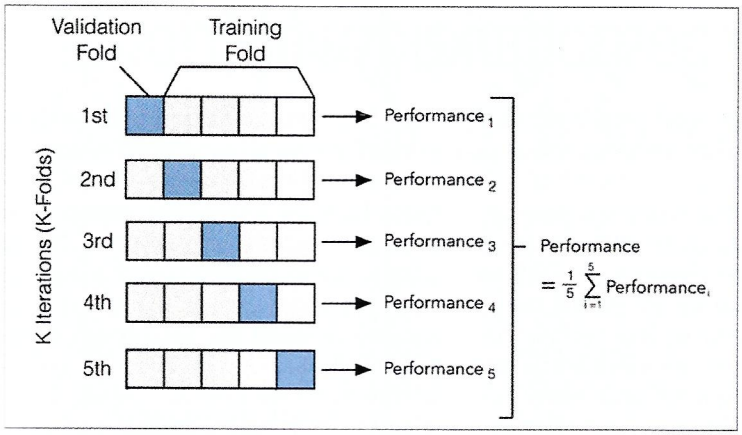
Istnieje wiele różnych metod radzenia sobie ze zjawiskiem przeuczenia modeli. Metody te zależą często od typu algorytmu oraz charakterystyki zjawiska. Bardzo dobre rezultaty daje zwiększenie liczebności próby treningowej, co jest możliwe, gdy posiadamy dużo danych. W procesie tworzenia modelu, w praktyce od razu wykorzystuje się wszystkie dane. Poza tym przeuczenie pojawia się często w sytuacji, gdy danych jest naprawdę mało. Podwojenie wielkiej ilości danych praktycznie nie może przynieść żadnego rezultatu.

Kolejną metodą jest upraszczanie algorytmu modelu, usztywnianie go lub zatrzymywanie w jakimś momencie procesu konwolucji. Te sposoby radzenia sobie z przeuczeniem wymagają dużej wiedzy i praktyki, ponieważ przeprowadzane są w sferze tzw. hiperparametrów modelu. Działanie takie najczęściej polega na metodzie zmieniania parametrów algorytmów na zasadzie

[1] Nie należy losowo dzielić zbiorów mających postać szeregów czasowych. W tym przypadku należy wydzielać okresy treningowe i okresy testowe.

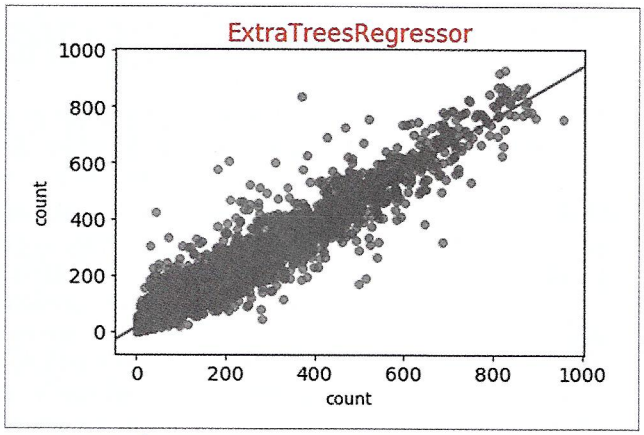


Wykres 1. Przykład zmiany hiperparametru gamma modelu SVM.



Wykres 2. Schemat działania walidacji krzyżowej.

Źródło: https://zitaoshen.rbind.io/project/machine_learning/machine-learning-101-cross--vaildation/



Wykres 3. Poziom dopasowania zmiennych empirycznych zbioru testowego do danych teoretycznych modelu Extra Trees.

prób i błędów, co samo w sobie jest bardzo czasochłonne i nie daje żadnych gwarancji powodzenia.

Kolejną metodą eliminacji przeuczenia modelu jest likwidacja lub dodawanie zmiennych opisujących (*feature elimination*). Prowadzi się weryfikację efektywności poszczególnych zmiennych, co jest procesem długim, skomplikowanym i obarczonym sporym ryzykiem popełnienia błędu. Metoda ta ma bardzo wiele wariantów ze względu na formę zmiennych niezależnych oraz zmiennych wynikowych.

Istnieje jednak metoda eliminacji przeuczenia, która jest uniwersalna i możliwa do zautomatyzowania. Metodą tą jest walidacja krzyżowa.

Walidacja krzyżowa

Metoda walidacji krzyżowej (*cross validation*) jest ściśle związana z nauką nadzorowaną. Jak pamiętamy w nauce z nadzorem musimy wydzielić z danych duży zbiór treningowy i mały testowy. Pamiętamy też, że przeuczenie to przede wszystkim słabe dopasowanie wyników empirycznych do uzyskanych przez model wyników teoretycznych, uzyskanych na podstawie danych testowych.

Metoda *cross validation* polega na dokonywaniu wielokrotnych podziałów na dane testowe i treningowe.

W naszym przykładzie dokonamy pięciu podziałów zbioru danych. Każdy podział nazywany jest *faudą*.

W pierwszej faudzie 20% to dane testowe, reszta – treningowe. Model trenowany jest na tym zbiorze testowym, następnie sprawdzany jest na danych testowych pierwszej faudy. W ten sposób model dowiaduje się jakie błędy popełnił w pierwszej faudzie. Następnie automatycznie tworzy się drugą faudę, gdzie wydziela się kolejne 20% danych do testowania. Nie mogą to być jednak te dane, które były w zbiorze testowym pierwszej faudy. Model powtarza proces treningu i testowania dla danych drugiej faudy. Odbywa się tyle cykli, ile zostało ich zadane w hiperparametrach narzędzia.

Tabela 1. Przykład wydruku diagnostycznego walidacji krzyżowej dla modelu Extra Trees dla regresji.

```
ExtraTreesRegressor
------------------------------cross_val, KFold = 9 ------
R2:  [0.92 0.92 0.92 0.91 0.94 0.92 0.91 0.92 0.92]
Mean_dev:  [-30.6 -30.6 -32.1 -34.3 -29.4 -33.3 -32.5 -31.9 -30.1]
```

Zastosowanie walidacji krzyżowej

Walidacja krzyżowa może być użyta jako narzędzie diagnostyczne, które wskaże na poziom jakości modelu; może być też metodą, która będzie automatycznie eliminowała przeuczenie.

W powyższym przykładzie badanie przeprowadzono na dziewięciu faudach. Pierwszy rząd oznacza parametr R^2, czyli współczynnik determinacji regresji dla zbioru testowego. Jak widać parametr ten jest dość wysoki i stabilny, co świadczy o wysokiej jakości modelu.

Drugi rząd tabeli oznacza statystykę odchylenia standardowego, który również jest stabilny dla wszystkich faud walidacji.

Podsumowanie

Obecnie technika *cross validation* jest najczęściej spotykaną w diagnostyce i optymalizacji modeli data science. Istnieje dużo gotowych bibliotek języka Python, które łatwo można zastosować w procesie badawczym.

Wojciech Moszczyński

Wojciech Moszczyński – absolwent Katedry Ekonometrii i Statystyki Uniwersytetu Mikołaja Kopernika w Toruniu, specjalista z zakresu ekonometrii, *data science* oraz rachunkowości zarządczej. Specjalizuje się w optymalizacji procesów produkcyjnych i logistycznych. Prowadzi badania w obszarze rozwoju i zastosowania sztucznej inteligencji. Od lat zaangażowany w popularyzację ekonometrii oraz *data science* w środowiskach biznesowych