Sprawozdanie - SI

Lab 4

Wiktor Sadowy

Cel zadania

W ramach zadania mieliśmy:

1. dokonać analizy danych
2. przygotować dane wykorzystując różne metody. W ramach zadania wykonaliśmy też zadanie bonusowe i przygotowaliśmy metody radzenia sobie z brakującymi danymi
3. przetestować klasyfikatory z użyciem różnych zestawów hiperparametrów. W ramach zadania wykonaliśmy też zadanie bonusowe i przetestowaliśmy bardziej zaawansowane algorytmy np. Las Losowy czy SVM
4. ocenić klasyfikację z pomocą różnych metryk oceny klasyfikacji i zinterpretować wyniki

Opis teoretyczny

W zadaniu zajmujemy się uczeniem nadzorowanym. Jest to sposób uczenia w którym zbiór danych treningowych zawiera dołączone rozwiązanie problemu tzw. etykiety albo klasy. Dzięki temu możemy rozwiązywać problem regresji (przewidywanie wartości) i problem klasyfikacji (przewidywanie klas).

W ramach zadania podzieliśmy musieliśmy podzielić nasz zbiór na dwie części:

1. Zbiór uczący – zestaw danych używany do nauki algorytmu
2. Zbiór walidacyjny – zestaw danych używany do przeprowadzania testów modelu. Testy są przeprowadzone po to, aby jak najlepiej dopasować hiperparametry modelu. Dzięki użyciu zbioru walidacyjnego jesteśmy w stanie sprawdzić czy model się generalizuje i poradzi sobie na nowych danych. Ważne jest to, żeby dane w zbiorze walidacyjnym nie były używane w zbiorze uczącym. W przeciwnym przypadku nie jest możliwa obiektywna ocena działania modelu

W problemach regresji i klasyfikacji często się też dodaje kolejny zbiór danych czyli zbiór testowy. Dane w tym zbiorze nie są używane ani w zbiorze uczącym ani zbiorze walidacyjnym i ich celem jest przetestowanie działania modelu na nowych danych i ocena czy prawidłowo wybraliśmy model i dopasowaliśmy do niego hiperparametry

Mogliśmy też zastosować walidację krzyżową, która jest alternatywnym sposobem przygotowania zestawów treningowych i testowych dla modelu. Metoda ta polega na podzielenie zestawu danych na podzestawy i utworzeniu z nich zestawów danych walidacyjnych i treningowych.

Jest wiele rodzajów walidacji krzyżowej np. k-krotna walidacja krzyżowa, która polega na tym, że dzielimy próbki na k grup. Model jest trenowany na k-1 grupach i walidowany na jednej grupie. Po wykonaniu k treningów wyniki przeprowadzonych walidacji są uśredniane.

W ramach zadania wykorzystujemy następujące modele:

1. Naiwny klasyfikator Bayesa – prosty klasyfikator probabilistyczny, który naiwnie zakłada niezależność i tą samą istotność cech dla ustalonej etykiety klasy. Naiwny klasyfikator Bayesa opiera się na twierdzeniu Bayesa. Dużą zaletą algorytmu jest jego prostota.
2. Drzewo decyzyjne – jest to reprezentacja klasyfikatora w postaci drzewa wspomagającego proces decyzyjny. Drzewo jest budowane w oparciu o różne algorytmy i parametry (np. entropia lub przyrost wiedzy), które są wyliczane dla każdego nowego węzła drzewa decyzyjnego. Drzewa decyzyjne poprzez swoją strukturę są bardzo podatne na zjawisko nadmiernego dopasowania gdyż wraz ze wzrostem głębokości coraz lepiej dostosowują się do zbioru uczącego, ale równocześnie pogarsza się ich możliwość generalizacji przez co gorzej radzą sobie ze zbiorem testowym
3. Las losowy – jest to algorytm, który używa wiele drzew decyzyjnych. Każde drzewo jest trenowane na innej części danych. Wówczas mamy gwarancję, że będziemy mieli różne drzewa. Wynikiem predykcji lasu losowego jest najczęściej przewidywana klasa przez drzewa decyzyjne. Zaletą lasu losowego jest to, że wykorzystuje on silne strony drzew decyzyjnych, jednocześnie zmniejszając ryzyko nadmiernego dopasowania
4. Klasyfikator wektorów nośnych – jest to algorytm, który dzieli przestrzeń w którym są nasze dane za pomocą funkcji (może być to np. funkcja liniowa czy wielomianowa) i na podstawie podziału dokonuje predykcji klasy. W przypadku gdy mamy do czynienia z dwoma klasami wystarczy jedna funkcja do podziału przestrzeni. W przypadku gdy mamy do czynienia z większą liczbą klas trzeba utworzyć funkcję oddzielającą klasę od każdej innej klasy (czyli dla 3 klas będą to 3 funkcje, dla 4 będzie to 6 funkcji itd.). Zaletą algorytmu jest to, że działa on bardzo dobrze gdy możemy bezproblemowo odróżnić od siebie klasy w oparciu o atrybuty lub gdy liczba atrybutów jest większa od liczby próbek. Wadą modelu jest to, że jest on bardzo podatny na różnice w wielkościach atrybutów (np. jeżeli jeden atrybut mieści się w przedziale 1-3, a drugi w przedziale 100000-300000 to model może mieć problemy z dopasowaniem się do danych). Oznacza to, że trzeba przed trenowaniem przetworzyć dane używając np. standaryzacji.

Przykładowe zastosowanie

Uczenie maszynowe możemy użyć do rozwiązywania różnych problemów regresji lub klasyfikacji np. przewidzenie czy dany mail jest spamem czy nie. Dla danego przykładu stosowanie wytrenowanych modeli pozwala na oszczędzenie czasu, gdyż nie musimy wówczas ręcznie decydować czy dany mail jest spamem czy nie.

Implementacja

Podczas wykonywania zadania korzystaliśmy z Jupyter Notebook. Po kolei wykonywaliśmy kolejne zadania, które były na liście. Podczas implementacji naszym głównym celem było napisanie takiego kodu, który pozwalałby na łatwe testowanie kolejnych modeli.

Materiały dodatkowe

Do wykonania zadania została użyta instrukcja do zadania, dokumentacja scikit-learn - <https://scikit-learn.org/stable/index.html> oraz poniższe strony:

<https://www.makeuseof.com/fill-missing-data-with-pandas/>

<https://medium.com/synthesio-engineering/precision-accuracy-and-f1-score-for-multi-label-classification-34ac6bdfb404>

Opis wykorzystanych bibliotek

Do zadania zostały wykorzystane następujące wbudowane biblioteki:

1. copy – biblioteka pozwalająca na kopiowanie zmiennych

Oraz następujące dodatkowe biblioteki:

1. pandas – biblioteka pozwalająca na wczytanie pliku z danymi. Używając biblioteki jesteśmy w stanie przetwarzać dane zawarte w pliku np. wypełnić brakujące dane
2. matplotlib – biblioteka służąca do tworzenia wykresów
3. seaborn – biblioteka oparta o matplotlib pozwalają na tworzenie wykresów. W porównaniu do matplotlib tworzenie wykresów jest o wiele prostsze
4. sklearn – biblioteka pozwalająca na użycie metod związanych z uczeniem maszynowym

Analiza danych

TO DO

Jak przygotowane dane wpływają na predykcję

TO DO

Jak wybór hiperparametrów wpływa na jakość predykcji

TO DO

Podsumowanie