Jacek Wołoszyn

Wyszukiwanie obiektów o podobnych cechach w bazie danych z wykorzystaniem sztucznej inteligencji = Search for Objects with Similar Features in the Database Using Artificial Intelligence

Dydaktyka Informatyki 13, 153-158

2018

Dydaktyka Informatyki 13(2018)
ISSN 2083-3156; e-ISSN 2543-9847
DOI: 10.15584/di.2018.13.20
http://www.di.univ.rzeszow.pl

Uniwersytet Rzeszowski

Jacek WOŁOSZYN

Dr inż., Uniwersytet Technologiczno-Humanistyczny w Radomiu, Wydział Informatyki i Matematyki, Katedra Informatyki, ul. Malczewskiego 29, 26-600 Radom; e-mail: jacek@delta.pl

WYSZUKIWANIE OBIEKTÓW O PODOBNYCH CECHACH W BAZIE DANYCH Z WYKORZYSTANIEM SZTUCZNEJ INTELIGENCJI

SEARCH FOR OBJECTS WITH SIMILAR FEATURES IN THE DATABASE USING ARTIFICIAL INTELLIGENCE

Słowa kluczowe: baza danych, sztuczna inteligencja.
Keywords: databases, artificial intelligence.

Streszczenie

Artykuł ten przedstawia proces tworzenia systemu rekomendującego obiekty z pól rekordów bazy danych o zbliżonych cechach podobieństwa do wybranych cech obiektu wzorcowego. W budowie wykorzystano algorytm K najbliższych sąsiadów, który często jest wykorzystywany do tworzenia systemów klasyfikacji.

Abstract

This article presents the process of creating a system that recommends objects from database records fields with similar selected features similar to the features of a reference object. The K algorithm of the nearest neighbors was used in the construction, which is often used to create classification systems.

Wstęp

Sztuczna inteligencja w realnym świecie staje się coraz bardziej powszechna. Znajduje zastosowanie w nowoczesnym stylu życia, ułatwia codzienne czynności takie jak: wyszukiwanie wiadomości, identyfikacja biometryczna, rozpoznawanie tekstu, obrazu, sterowanie urządzeniami i wiele innych.

Organizacje na całym świecie używają zgromadzonych danych do rozwiązywania krytycznych problemów biznesowych. Znajomość algorytmów, narzę-

dzi statystycznych, problemu i obsługi danych historycznych jest niezbędna do rozwiązania zagadnienia za pomocą modelowania.

Zadanie zrealizowano w języku programowania Python3, z wykorzystaniem bibliotek Numpy, Pandas, Matplotlib, Scikit-learn zapewniający modelowanie.

Python jest rozwijany jako projekt Open Source. Zapewnia wraz z bibliotekami szybkie dojrzałe środowisko do analizy danych, nauczania maszynowego i szeroko pojętej analizy danych.

Celem pracy jest opracowanie algorytmu pozwalającego na wyszukanie obiektów o podobnych właściwościach w przestrzeni bazy danych.

Opis problemu

Modelowanie[1] zagadnień jest sztuką interdyscyplinarną i wymaga znajomości czterech dyscyplin, statystyki, algorytmów, narzędzi i wiedzy eksperckiej modelowanej dyscypliny. Modelowanie można zapisać w postaci równoważnej macierzy 2x2 (rys. 1).

MODELOWANIE
[statystyka algorytmy]
[narzędzia wiedza]

Rys. 1. Macierz dyscyplin modelowania

Przyjęto założenie, że system informatyczny, zapisuje do bazy danych[2] rekordy zawierające informacje o osobach. Każdy zapisany rekord zawiera informację o obiekcie (w naszym przypadku o osobie). W rekordzie oprócz danych identyfikujących jednoznacznie obiekt (w tym przypadku imię i nazwisko osoby) występują pola opisujące zainteresowania, upodobania, cechy obiektu.

Na potrzeby tego artykułu wygenerowano losowe dane zawierające pola rekordu, w tym pola cecha_A oraz cecha_B. Zakładamy, że wartości te są wpisane przez użytkownika bezpośrednio bądź też obliczone za pomocą odpowiedniego algorytmu uwzględniające wagę cech. Definiują one jednoznacznie preferencję użytkownika.

Mogą się pod nimi kryć jego upodobania do oglądanych filmów, do pożądanych cech u drugiej osoby, zainteresowania, preferencje podróży lub inne.

Na listingu 1, przedstawiono przykładowe rekordy identyfikujące użytkownika, jak i jego upodobania.

[1] A. Boschetti, L. Massaron, *Python. Podstawy nauki o danych*, Gliwice 2017.

[2] M. Goodrich, R. Tamassia, M. Goldwasser, *Data Structures and Algorithms in Python*, Wiley 2013.

	Cecha_A	Cecha_B	Hobby	Płeć	Waga	Wiek	Wzrost
0	19.91	18.84	Czasami	Kobieta	65.06	28.70	146.75
1	16.76	18.82	Nie	Kobieta	50.08	24.64	183.71
2	20.77	22.05	Tak	Mężczyzna	52.14	36.57	104.34
3	18.30	17.72	Czasami	Mężczyzna	33.40	25.10	136.98
4	22.99	21.19	Nie	Kobieta	32.66	55.06	146.16

Listing 1. Fragment materiału źródłowego

Dla dalszych rozważań wykorzystywane będą pola rekordu cecha_A oraz cecha_B.

Budowa algorytmu

Rozwiązanie problemu przedstawia się następująco. Użytkownik poszukuje obiektu/obiektów osoby/osób o najbardziej zbliżonych cechach do swoich upodobań. Oznacza to, że wpisuje oczekiwane wartości cech listing 4 i w odpowiedzi otrzymuje obiekty o najbardziej zbliżonych wartościach. Na przykład użytkownik otrzymuje propozycję listy filmów, charakteryzujące się tym, że są maksymalnie zbliżone do jego oczekiwań lub studentów o zbliżonych umiejętnościach listing 4.

Z pól rekordu należy wykorzystać wspomniane już wcześniej pola cecha_A i cecha_B, których wartość została wcześniej ustalona za pomocą odpowiedniego algorytmu wspólnego dla wszystkich. Aby pobrać pola została użyta ramka danych DataFrames z biblioteki Pandas[3], do której zostały wybrane wcześniej rekordy listing 2.

```
X = df[['Cecha_A', 'Cecha_B']].values
```

Listing 2. Utworzona macierz typu {ndarray} ze zbioru rekordów

Df to obiekt typu {dataframe} do, którego zostały wczytane rekordy z bazy danych pokazane na listingu 1. Natomiast X jest macierzą zawierającą już wyselekcjonowane cechy. Użycie values jest konieczne dla zmiany struktury pozyskanych danych z postaci {DataFrame} na {ndarray}. Brak tej zamiany uniemożliwi dalsze przetwarzanie danych z wykorzystaniem algorytmu w obecnej postaci.

```
[[19.91933944 18.14771215…….]]
```

Listing 3. Fragment otrzymanej macierzy X

[3] Y. Hilpisch, *Derivatives Analytics with Python*, Wiley 2015.

Kolejnym krokiem jest ustalenie cech, jakich oczekujemy w poszukiwanych obiektach. Cechy należy za pomocą tego samego algorytmu przetworzyć do postaci liczbowej tak jak wcześniej przetworzono cechy w rekordach bazy danych. Tak uzyskane wartości cech zapisujemy do tablicy z nazwą datapoint.

```
Datapoint = [22.3, 16.7]
```

Listing 4. Lista danych wzorcowych

Posiadając źródłowy zestaw informacji można przystąpić do wyznaczania najbliższych sąsiadów dla punku datapoint. Wykorzystana zostanie funkcja Nearest Neighbors[4] z biblioteki sklearn. Zapewnia ona funkcjonalność dla nadzorowanych i nienadzorowanych metod uczenia się. Główną zasadą leżącą u podstaw metod algorytmu jest odnalezienie predefiniowanej liczby próbek treningowych położonych najbliżej zdefiniowanego punktu. Przyjmuje ona parametry n_neighbors, algorithm, leaf_size, metric, p, metric_params, n_jobs.

n_neighbors – domyślnie przyjmuje wartość 5. Oznacza, ile najbliższych obiektów zostanie wyszukanych,

algorithm – algorytm: {'auto', 'ball_tree', 'kd_tree', 'brute'}

Dokładny opis wszystkich możliwych parametrów można uzyskać w opisie biblioteki.

```
      knn_model = NearestNeighbors(n_neighbors=k,    algo-
      rithm='ball_tree').fit(X)
distances, indices = knn_model.kneighbors([datapoint])
```

Listing 5. Tworzenie i treninig modelu K-najbliższych sąsiadów

W wyniku wykonanego polecenia w zmiennej distances typu {ndarray} iindices typu {ndarray} przechowywane są wyniki obliczeń zawierające pięć najbliższych punków sąsiadów względem zadanego punktu wzorcowego.

Kolejnym krokiem jest wyprowadzenie wartości ze zmiennej do postaci czytelnej dla użytkownika, jak i wizualizacja uzyskanych wyników.

```
print("\nOsoby o najbliższych pożądanych cechach:")
for rank, index in enumerate(indices[0][:k], start=1):
print(str(rank) + " osoba >>>", X[index])
plt.figure()
plt.title('Najbliżsi sąsiedzi')
plt.scatter(X[:, 0], X[:, 1], marker='o', s=75, color='k')
plt.scatter(X[indices][0][:][:, 0], X[indices][0][:][:, 1],
marker='o', s=250, color='red', facecolors='none')
```

[4] S. Raschka, V. Mirjalili, *Python Machine Learning Packt 2017*.

```
plt.scatter(test_datapoint[0], test_datapoint[1],
marker='x', s=75, color='k')
plt.show()
```

Listing 6. Prezentacja wyników

Wynik działania algorytmu

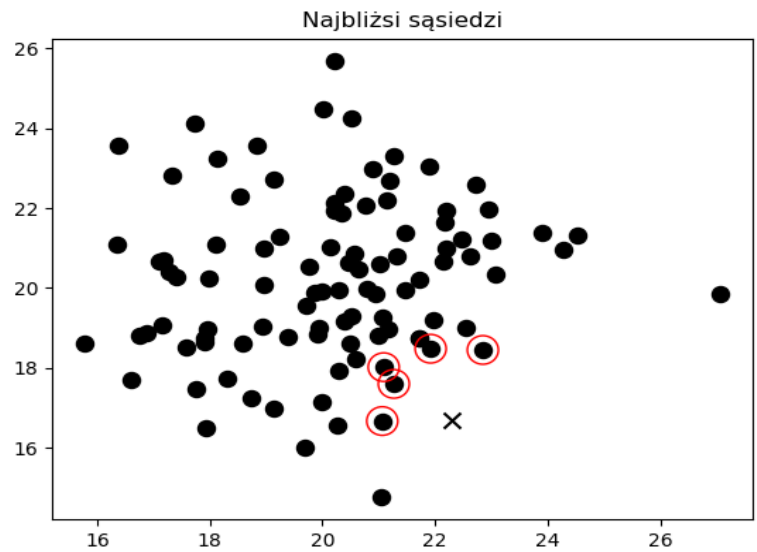
Użyte w algorytmie dane same w sobie nie przedstawiają wielkiej wartości. To zwykłe rekordy zawierające pola. Jednak przetworzenie tych danych za pomocą prostego algorytmu pozwala uzyskać użyteczne informacje.

Na listingu 7 zostały wydrukowane wyniki poszukiwania najbliższych pięciu sąsiadów do wzorca. Rys. 2 przedstawia interpretację graficzną rozwiązania. Czarne koła reprezentują wartości cech obiektów, x reprezentuje wzorzec do wartości, którego są wyznaczone.

Osoby o najbliższych pożądanych cechach – czarny okrąg:

```
1 osoba >>> [ 21.0620175   16.66772775]
2 osoba >>> [ 21.26730482  17.597991  ]
3 osoba >>> [ 21.08773664  18.01741686]
4 osoba >>> [ 21.921       18.47942694]
5 osoba >>> [ 22.84843486  18.44806887]
```

Listing 7. Uzyskane rezultaty



Rys. 2. Wynik działania algorytmu – postać graficzna

Pamiętając o fakcie, że w bazie rekordy przechowują również dane bezpośrednio identyfikujące obiekt/osobę możemy bezpośrednio wskazać na obiekty o poszukiwanych cechach. Wybrany algorytm może znaleźć zastosowanie np. w serwisach kojarzących pary lub wyszukujących podróże uwzględniając upodobania klientów. Kluczem do prawidłowego działania serwisu nie jest tu jednak tylko sam algorytm, ale umiejętne powiązanie wartości cech, które zamierzamy wyszukać.

Podsumowanie

Ilość generowanych danych we współczesnym świecie rośnie w szybkim tempie. Dane są tworzone przy okazji zakupów w Amazon, przy okazji płatności w PayPal, generowania raportów, drukowania biletów, rezerwacji miejsc, monitoringu miejskiego. Same media społecznościowe generują terabajty danych. Znajdują się tam informacje, gdzie mieszkamy, skąd pochodzimy, co lubimy, co kupujemy, ile wydajemy pieniędzy, dokąd podróżujemy lub dokąd zamierzamy pojechać. Ten zbiór danych zawiera wiele informacji. Odpowiednio napisane algorytmy mogą pomóc w podejmowaniu decyzji, zasugerować pewne rozwiązania. Wszystkie wyniki opierają się na rzeczywistych danych, które sami zgromadziliśmy. Ta wiedza może być użyta, żeby nam pomóc w codziennym życiu, ale nie tylko. Algorytmy z wykorzystaniem sztucznej inteligencji wykorzystuje policja, aby przewidzieć zdarzenia z wyprzedzeniem na podstawie danych historycznych, ale wykorzystują także organizacje przestępcze, aby zminimalizować ryzyko swojej działalności.

Bibliografia

Boschetti A., Massaron L., *Python. Podstawy nauki o danych*, Helion, Gliwice 2017.
Goodrich M., Tamassia R., Goldwasser M., *Data Structures and Algorithms in Python*, Wiley 2013.
Hilpisch Y., *Derivatives Analytics with Python*, Wiley 2015.
Raschka S., Mirjalili V., *Python Machine Learning Packt 2017*.