234053
Numer indeksu
Paweł Galewicz
Imię i nazwisko

234067
Numer indeksu
Bartosz Jurczewski
Imię i nazwisko

234102
Numer indeksu
Zbigniew Nowacki
Imię i nazwisko

234106
Numer indeksu
Karol Podlewski
Imię i nazwisko

234128
Numer indeksu
Piotr Wardęcki
Imię i nazwisko

Kierunek Informatyka Stosowana
Stopień II
Specjalizacja Data Science
Semestr 1

Data oddania 18 marca 2020

Metody uczenia maszynowego

Problem set 1

Spis treści

1 Cel

Zadanie polegało na analizie procesu klasyfikacji danych za pomocą wybranych metod:

1. Algorytm drzew decyzyjnych
2. Naiwny klasyfikator Bayesa
3. Maszyna wektorów nośnych
4. Klasyfikator k-najbliższych sąsiadów
5. Algorytm sztucznych sieci neuronowych

Należało zaimplementować każdą metodę, a następnie zweryfikować jej działanie biorąc pod uwagę:

- rożne możliwe ustawienia parametrów konfiguracyjnych i ich wpływ na wyniki klasyfikacji
- zbiory danych o różnej charakterystyce (przynajmniej 3 różne zbiory)

Każdą metodę należało przetestować na tych samych zbiorach, a następnie porównać wyniki i wyciągnąć wnioski dotyczące skuteczności poszczególnych metod. Jako kryterium porównawcze wykorzystaliśmy dokładność klasyfikacji.

2 Opis implementacji

Algorytmy zostały zaimplementowane za pomocą języka Python w wersji 3.8.2. Wykorzystano w nim biblioteki NumPy, Sklearn i Pandas.

3 Opis zbiorów danych

Bazowaliśmy na trzech zestawach danych:

A. Fall Detection Data from China - zbiór składający się z 6 cech – czasu oraz pięciu parametrów życiowych. Klasyfikowany jest stan pacjenta w danym przypadku – stanie, chód, siedzenie, upadek, kurcz, bieg (łącznie 6).

B. Rain in Australia – historia danych pogodowych z 10 lat (data, lokalizacja, temperatury, opady, wiatr, ciśnienie, wilgotność, nasłonecznienie itp). Na podstawie 23 cech należy określić, czy kolejnego dnia wystąpią opady.

C. Pima Indians Diabetes Database – zbiór danych zawierający 8 niepowiązanych ze sobą pomiarów diagnostycznych (m. in. ilość ciąż, poziom glukozy, ciśnienie, bmi, wiek) na podstawie których dokonywana jest ocena, czy kobieta choruje na cukrzycę.

4 Klasyfikatory

4.1 Algorytm drzew decyzyjnych

Algorytm polega na stworzeniu modelu do przewidywania wartości na podstawie prostych reguł wywnioskowanych z danych treningowych. Reguły te tworzone są w struktury drzewiaste. Struktury te składają się z:

- węzła głównego – od niego rozpoczyna się proces decyzyjny
- węzłów decyzyjnych – zawierające reguły-zapytania
- stanów (liścia) – końcowych stanów algorytmu, w problemie klasyfikacji są one równoważne z etykietami
- połączeń między węzłami – reprezentującymi możliwe warianty dla danego

Zapytania w węzłach są wyrażeniami logicznymi dotyczącymi jednej z cech modelu oraz jej wartości. Wartość ta dobrana musi być w taki sposób, żeby jak najlepiej wydzielić klasę obiektów z przychodzących na węźle danych. Można wtedy powiedzieć, że dany węzeł dostarcza najwięcej informacji. Na potrzebę obliczenia tego przyrostu informacji wprowadza się kryterium *Ipurity*, którego sensem jest fakt, czy po podziale w danym węźle dane zostały poprawnie klasyfikowane. Dokładny sposób wyliczania wartości tego kryterium jest zależny od konfiguracji.

Drzewa domyślnie budowane są do momentu zminimalizowania wartości Impurity, przez co struktura drzew może być bardzo złożona. Skutkiem tego może być przeuczenie modelu, co rzutuje na jego dokładność. Aby ograniczyć możliwość wystąpienia tego zjawiska wprowadza się dodatkowy parametr – *maksymalna głębokość* – który mówi o tym ile najwięcej rozgałęzień może wystąpić między węzłem głównym a liściem.

4.2 Naiwny klasyfikator Bayesa

Naiwny klasyfikator Bayesa dokonuje klasyfikacji na bazie twierdzenia Bayesa:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

gdzie:

- A, B – zdarzenia
- $P(A \mid B)$ – prawdopodobieństwo zdarzenia A, o ile zajdzie B
- $P(B \mid A)$ – prawdopodobieństwo zdarzenia B, o ile zajdzie A
- $P(A)$ – prawdopodobieństwo wystąpienia zdarzenia A
- $P(B)$ – suma prawdopodobieństw wszystkich potencjalnych skutków zdarzenia: $P(B) = \sum P(B \mid A)P(B)$

Model naiwnego klasyfikatora Bayesa zakłada, że dana cecha klasy jest niepowiązana z pozostałymi cechami. Każda z cech indywidualnie wskazuje na prawdopodobieństwo przynależności do danej klasy. Sprawdza się najlepiej przy dużych zbiorach danych. Jest wykorzystywany m.in. przy filtrowaniu spamu, diagnozie medycznej, czy prognozowaniu pogody.

4.3 Maszyna wektorów nośnych

Maszyna wektorów nośnych jest klasyfikatorem liniowym. Algorytm polega na rozdzieleniu obiektów o różnej przynależności klasowej za pomocą hiperpłaszczyzn, które mają być od siebie możliwe jak najbardziej oddalone - taką odległość nazywa się marginesem klasyfikatora, a hiperpłaszczyzny z największym marginesem wektorami nośnymi.

Algorytm bardzo dobrze sobie radzi z danymi liniowo separowanymi, ale nie zawsze będzie istniała hiperpłaszczyzna rozdzielająca, która zapewni poprawną klasyfikację wszystkich elementów zbioru. W takich przypadkach maszyna wektorów nośnych za pomocą funkcji jądrowych transformuje przestrzeń do postaci liniowo separowanej.

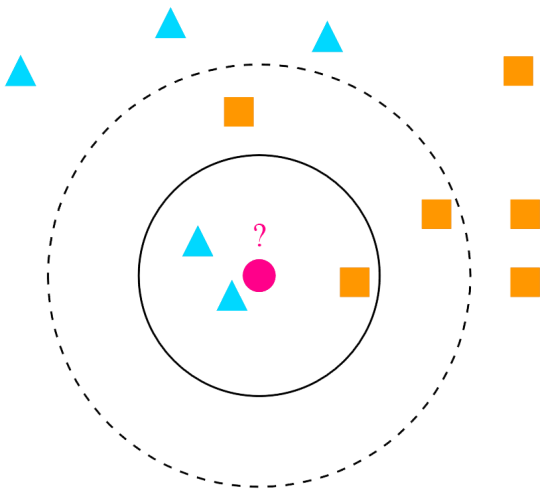
Algorytm umożliwia zmianę wielu parametrów, przy czym najpopularniejsze z nich to:

- Regularyzacja - parametr określający jak rygorystyczna jest klasyfikacja. Im większa wartość parametru, tym większy margines ma być znaleziony. Domyślna wartość to 1.
- Jądro - typ funkcji przekształcającej problem nierozdzielny do problemu rozdzielnego, na przykład funkcja liniowa, radialna (domyślna), wielomianowa czy sigmoidalna.
- Gamma - określa jak duży wpływ pojedynczy trening ma na inne treningi - im wyższa wartość gammy, tym coraz bardziej podobne muszą być do siebie kolejne próby, by były brane pod uwagę. Ma zastosowanie tylko dla funkcji jądra radialnej, wielomianowej i sigmoidalnej.

4.4 Klasyfikator k-najbliższych sąsiadów

Algorytm k najbliższych sąsiadów jest klasyfikatorem (ściślej algorytmem regresji regresji nieparametrycznej). Algorytm ten zakłada dany zbiór uczący, w którym znajdują się już sklasyfikowane dane. Schemat składa się z szukania k obiektów najbliższych do obiektu klasyfikowanego. Następnie, przyporządkowuje się nowy obiekt do najczęściej występującej klasy w obrębie jego k-najbliższych sąsiadów.

Rysunek 1 pokazuje działanie algorytmu. W przypadku k=3 (mniejszy okrąg), różowa kropka zostanie zakwalifikowana do niebieskich trójkątów. W przypadku k=5 (większy okrąg) - do pomarańczowych kwadratów.



Rysunek 1: Wizualizacja klasyfikatora k-najbliższych sąsiadów

4.5 Algorytm sztucznych sieci neuronowych

Sztuczna siec neuronowa jest połączeniem wielu elementów nazywanych sztucznymi neuronami, które tworzą co najmniej trzy warstwy: wejściową, ukrytą oraz wyjściową. Neurony przetwarzają informacje dzięki nadaniu im parametrów które nazywane są wagami. Podstawą tworzenia sieci neuronowej jest modyfikowanie współczynnika wagowego połączeń w celu uzyskania poprawnych wyników. W programie użyty został algorytm MLP (ang. Multilayer Perceptron).

5 Badania

W sekcji badań odnoszono się do zbiorów danych za pomocą oznaczeń zgodnych z sekcją 3.

5.1 Algorytm drzew decyzyjnych

Wielkość zbioru treningowego

Na początku sprawdzono wpływ podziału zbioru na części treningowe oraz testowe przy domyślnych cechach klasyfikatora.

% danych treningowych	Zbiór danych		
	A	B	C
60	0.678	1.0	0.668
65	0.681	1.0	0.717
70	0.677	1.0	0.711
75	0.687	1.0	0.739
80	0.693	1.0	0.680
85	0.697	1.0	0.703
90	0.690	1.0	0.699

Tabela 1: Porównanie dokładności dla różnych zbiorów, dla drzew decyzyjnych

Dla 75% danych testowych dokładność klasyfikacji była nieznacznie lepsza, jednak podział nie miał znaczącego wpływu na ogólne wyniki. Wspomniany podział został wykorzystany przy kolejnych badaniach.

Zmiana kryterium Impurity

Kolejnym badaniem było porównanie wpływu kryterium Impurity, które wykorzystywane jest przy oblicaniu przyrostu informacji.

	Gini	Entropy
Dataset A	0.696	0.711
Dataset B	1.0	1.0
Dataset C	0.692	0.691

Tabela 2: Porównanie dokładności dla różnych kryteriów Impurity algorytmu drzew decyzyjnych

Zmiana maksymalnej głębokości

Następnie sprawdzono wpływ maksymalnej głębokości. Ustawienie jej może ograniczyć zjawisko przeuczenia się modelu.

Maksymalna głębokość	Zbiór danych		
	A	B	C
2	0.363	1.0	0.710
4	0.408	1.0	0.721
6	0.477	1.0	0.735
8	0.565	1.0	0.712
10	0.640	1.0	0.705
12	0.674	1.0	0.697
14	0.683	1.0	0.699
16	0.691	1.0	0.711
18	0.693	1.0	0.696

Tabela 3: Porównanie dokładności dla różnej maksymalnej głębokości algorytmu drzew decyzyjnych

Zmiana maksymalnej liczby liści

Ostatnim eksperymentem było sprawdzenie zależności między maksymalną liczbą liści drzewa, a dokładnością klasyfikacji. Założono, że minimalna liczba liści musi być równa liczbie klas zestawu danych, następnie liczbę tę zwiększano o wielokrotność tej liczby.

Maksymalna liczba liści	**Zbiór danych**		
	A	**B**	**C**
1x liczba klas	0.378	1.0	0.705
2x liczba klas	0.413	1.0	0.727
4x liczba klas	0.464	1.0	0.738
6x liczba klas	0.506	1.0	0.736
8x liczba klas	0.534	1.0	0.739
10x liczba klas	0.566	1.0	0.727

Tabela 4: Porównanie dokładności dla maksymalnej liczby liści drzewa decyzyjnego

5.2 Naiwny klasyfikator Bayesa

Przy klasyfikatorze Bayesa nie mamy do dyspozycji parametrów innych niż podział na dane treningowe i testowe. Dokładność klasyfikacji sprawdzono dla każdego ze zbiorów danych w zależności od podziału.

% danych treningowych	**Zbiór danych**		
	A	**B**	**C**
60 %	0.131	0.940	0.739
65 %	0.139	0.941	0.761
70 %	0.132	0.942	0.761
75 %	0.135	0.941	0.775
80 %	0.134	0.942	0.712
85 %	0.138	0.939	0.730
90 %	0.135	0.946	0.737

Tabela 5: Porównanie dokładności dla różnych zbiorów, dla naiwnego klasyfikatora Bayesa

Dokładność klasyfikacji zależy przede wszystkim od samego zbioru danych. Natomiast jego podział ma pomijalny wpływ na wyniki.

5.3 Maszyna wektorów nośnych

Wielkość zbioru treningowego

Na początku sprawdzono wpływ podziału zbioru na części treningowe oraz testowe przy domyślnych cechach klasyfikatora.

% danych treningowych	Zbiór danych		
	A	**B**	**C**
60 %	0.3	0.803	0.762
65 %	0.317	0.809	0.761
70 %	0.326	0.813	0.757
75 %	0.341	0.818	0.78
80 %	0.33	0.818	0.725
85 %	0.322	0.817	0.73
90 %	0.324	0.819	0.737

Tabela 6: Porównanie dokładności dla różnych zbiorów, dla maszyny wektorów nośnych

Podział zbioru nie wpływa w znaczącym stopniu na dokładność klasyfikacji. Średnio najlepsze wyniki uzyskano dla zbioru treningowego stanowiącego 75%, i to właśnie taki podział będzie stosowany przy analizie wpływu zmiany parametrów klasyfikatora.

Zmiana parametru regularyzacji oraz funkcji jądra

Następnie porównano wpływ zmiany parametru regularyzacji oraz funkcji jądra na uzyskane wyniki.

Liniowa funkcja jądra nie była w stanie w ciągu 10 minut przeprowadzić klasyfikacji zbioru A, dlatego nie została ona uwzględniona w wynikach.

Parametr regularyzacji	Funkcja jądrowa			
	Liniowa	**Radialna**	**Wielomianowa**	**Sigmoidalna**
0.1	-	0.286	0.275	0.211
0.2	-	0.294	0.271	0.259
0.5	-	0.293	0.292	0.261
1	-	0.324	0.295	0.286
2	-	0.353	0.29	0.277
5	-	0.393	0.294	0.257

Tabela 7: Porównanie dokładności dla zbioru A przy zmianie regularyzacji oraz funkcji jądra

Parametr regularyzacji	Funkcja jądrowa			
	Liniowa	**Radialna**	**Wielomianowa**	**Sigmoidalna**
0.1	1.0	0.784	0.779	0.658
0.2	1.0	0.774	0.788	0.655
0.5	1.0	0.786	0.807	0.649
1	1.0	0.816	0.826	0.654
2	1.0	0.85	0.846	0.654
5	1.0	0.87	0.88	0.668

Tabela 8: Porównanie dokładności dla zbioru B przy zmianie regularyzacji oraz funkcji jądra

Parametr regularyzacji	Funkcja jądrowa			
	Liniowa	Radialna	Wielomianowa	Sigmoidalna
0.1	0.832	0.702	0.791	0.654
0.2	0.733	0.681	0.733	0.675
0.5	0.801	0.738	0.801	0.455
1	0.759	0.791	0.728	0.545
2	0.796	0.696	0.738	0.487
5	0.77	0.702	0.785	0.712

Tabela 9: Porównanie dokładności dla zbioru C przy zmianie regularyzacji oraz funkcji jądra

Zmiana gammy

Na sam koniec porównano zmianę gammy dla funkcji radialnej na zbiorze danych C dla parametru regularyzacji równego 1, mając na celu sprawdzenie, czy dzięki temu będą ona mogła się zbliżyć do dokładności klasyfikacji osiągniętych przez funkcję liniową czy wielomianową.

Parametr gammy	Funkcja radialna
0.1	0.654
0.2	0.634
0.5	0.649
1	0.67
2	0.702
5	0.654

Tabela 10: Porównanie dokładności dla zbioru C przy zmianie gammy dla funkcji jądrowej radialnej

5.4 Klasyfikator k-najbliższych sąsiadów

Na początku sprawdzono wpływ podziału zbioru na części treningowe oraz testowe przy domyślnych cechach klasyfikatora.

% danych treningowych	Zbiór danych		
	A	B	C
60%	0.638	0.858	0.713
65%	0.646	0.862	0.664
70%	0.65	0.859	0.752
75%	0.651	0.866	0.738
80%	0.652	0.859	0.719
85%	0.644	0.872	0.722
90%	0.668	0.859	0.658

Tabela 11: Porównanie dokładności dla różnych zbiorów, dla klasyfikatora k-najbliższych sąsiadów

Podział zbioru nie wpływa w znaczącym stopniu na dokładność klasyfikacji. Średnio najlepsze wyniki uzyskano dla zbioru treningowego stanowiącego 85%, i to właśnie taki podział będzie stosowany przy analizie wpływu zmiany parametrów klasyfikatora.

Liczba sąsiadów	Zbiór danych		
	A	B	C
2	0.669	0.853	0.713
3	0.670	0.857	0.678
4	0.670	0.859	0.687
5	0.659	0.868	0.791
6	0.656	0.866	0.774

Tabela 12: Porównanie dokładności dla różnych zbiorów klasyfikatora k-najbliższych sąsiadów, przy zmianie liczby sąsiadów

5.5 Algorytm sztucznych sieci neuronowych

Dla poszczególnych zbiorów danych najlepsze wyniki uzyskano dla 70 % danych treningowych. Następnie przystąpiono do badania wpływu zmian parametrów na klasyfikator. W tym celu zmieniana była liczba warstw ukrytych w algorytmie.

% danych treningowych	Zbiór danych		
	A	B	C
60 %	0.281	0.997	0.664
65 %	0.284	0.990	0.683
70 %	0.285	0.993	0.700
75 %	0.275	0.992	0.665
80 %	0.281	0.965	0.693
85 %	0.278	0.988	0.617
90 %	0.280	0.994	0.684

Tabela 13: Porównanie dokładności dla różnych zbiorów, dla sztucznych sieci neuronowych

Liczba warstw ukrytych	Zbiór danych		
	A	B	C
50	0.232	0.882	0.691
100	0.285	0.993	0.700
200	0.275	0.983	0.730
300	0.287	0.782	0.652
400	0.285	0.995	0.713
500	0.288	0.978	0.717

Tabela 14: Porównanie dokładności dla różnych zbiorów, dla 70 % danych treningowych oraz różnej liczby warstw ukrytych

6 Wnioski

Drzewa decyzyjne

- Zbiór danych mówiący o deszczach nie był dobrze dobrany do wykonywania testów tego algorytmu.
- Oba rodzaje kryterium Impurity dostarczały podobnych rezultatów.
- Do poprawnej klasyfikacji wymagana jest odpowiednio dobrana do zestawu danych maksymalna głębokość. Jak widać w tabeli 3, dla zbioru o wielu klasach jest ona zdecydowanie większa niż dla zbioru z dwoma klasami. Dla zbioru C zauważalny jest efekt przeuczenia.

- Podobnie jest w przypadku liczby liści, przy czym ich liczba równa liczbie klas zbioru okazuje się być zdecydowanie za mała.

Naiwny klasyfikator Bayesa

- Większa liczba cech, a mniejsza klas pozwala na dokładniejszą klasyfikację
- Rozmiar zbioru treningowego ma pomijalny wpływ na dokładność klasyfikacji

Maszyna wektorów nośnych

- Maszyna wektorów nośnych nie radzi sobie dobrze z klasyfikacją zbioru danych dotyczącego upadków ludzi w chinach (zbiór A) - jest to prawdopodobnie spowodowane większą liczbą klas do odróżnienia, kiedy algorytm najlepiej sobie radzi przy dwóch klasach. Zbiór ten był najdłuższej oraz najgorzej klasyfikowany.
- Parametr regularyzacji najlepiej się sprawdzał, kiedy jego wartość oscylowała w pobliżu wartości domyślnej równej 1.
- Najlepszą funkcją jadra przy klasyfikacji zbiorów opisanych w sekcji 3 okazała się funkcja liniowa. Bardzo dobre wyniki osiągały też funkcje radialna oraz wielomianowa. Dla sprawdzanych zbiorów danych zdecydowanie najgorsza okazała się funkcja sigmoidalna.
- Zmiana gammy nie wpłynęła zauważalnie na wyniki klasyfikacji na zbiorze dotyczącym cukrzyków.

Klasyfikator k-najbliższych sąsiadów

- Klasyfikator najlepsze wyniki osiąga na dużych zbiorach danych (zbiór B) w których to dane są ze sobą mocno skorelowane.
- Zmiana liczby sąsiadów nie miała dużego przełożenia na współczynnik dokładności.

Sztuczna sieć neuronowa

- Sztuczna sieć neuronowa najlepsze wyniki osiąga na dużych zbiorach danych (zbiór B) w których to dane są ze sobą mocno skorelowane.
- Średnio najlepsze wyniki dla wszystkich zbiorów danych uzyskiwane były dla 400 warstw ukrytych