

<u>234053</u>
<i>Numer indeksu</i>
<u>Paweł Galewicz</u>
<i>Imię i nazwisko</i>

<u>234067</u>
<i>Numer indeksu</i>
<u>Bartosz Jurczewski</u>
<i>Imię i nazwisko</i>

<u>234102</u>
<i>Numer indeksu</i>
<u>Zbigniew Nowacki</u>
<i>Imię i nazwisko</i>

<u>234106</u>
<i>Numer indeksu</i>
<u>Karol Podlewski</u>
<i>Imię i nazwisko</i>

<u>234128</u>
<i>Numer indeksu</i>
<u>Piotr Wardecki</u>
<i>Imię i nazwisko</i>

<b>Kierunek</b>	Informatyka Stosowana
<b>Stopień</b>	II
<b>Specjalizacja</b>	Data Science
<b>Semestr</b>	1

**Data oddania** 18 marca 2020

## Metody uczenia maszynowego

### Problem set 1

# Spis treści

<b>1</b>	<b>Cel</b>	<b>3</b>
<b>2</b>	<b>Opis implementacji</b>	<b>3</b>
<b>3</b>	<b>Klasyfikatory</b>	<b>3</b>
3.1	Algorytm drzew decyzyjnych . . . . .	3
3.2	Naiwny klasyfikator Bayesa . . . . .	4
3.3	Maszyna wektorów nośnych . . . . .	4
3.4	Klasyfikator k-najbliższych sąsiadów . . . . .	4
3.5	Algorytm sztucznych sieci neuronowych . . . . .	5
<b>4</b>	<b>Badania</b>	<b>5</b>
4.1	Fall Detection Data from China . . . . .	6
4.2	Rain in Australia . . . . .	6
4.3	Suicide Rates Overview 1985 to 2016 . . . . .	6
<b>5</b>	<b>Wnioski</b>	<b>6</b>

# 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:

- różne 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 (accuracy) oraz ...**.

## 2 Opis implementacji

Algorytmy zostały zaimplementowane za pomocą języka Python w wersji 3.8.2. Wykorzystano w nim biblioteki NumPy, Sklearn i Pandas. Bazowaliśmy na trzech zestawach danych:

- [Fall Detection Data from China](#)
- [Rain in Australia](#)
- [Suicide Rates Overview 1985 to 2016](#)

## 3 Klasyfikatory

### 3.1 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 węzele 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.

### 3.2 Naiwny klasyfikator Bayesa

Naiwny klasyfikator Bayesa dokonuje klasyfikacji na bazie twierdzenia Bayesa:

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)}$$

gdzie:

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

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.

### 3.3 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 hiperpłaszczyzn z największym marginesem wektorami nośnymi.

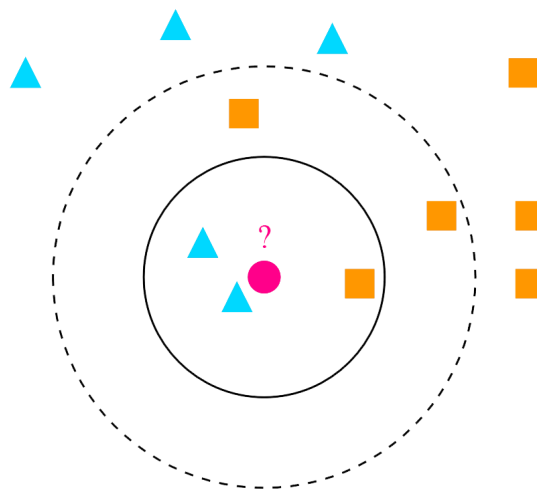
Algorytm bardzo dobrze sobie radzi z danymi liniowo separowanymi, ale nie zawsze będzie istniała hiperpłaszczyzna rozdzielająca, która zapewni poprawną klasyfikację wszystkich elementów zbioru. W takich przypadkach maszyna wektorów nośnych za pomocą funkcji jądrowych transformuje przestrzeń do postaci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

### 3.4 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:

### 3.5 Algorytm sztucznych sieci neuronowych

Sztuczna sieć 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.

## 4 Badania

W tabelach 1, 2, 3 zaprezentowano porównanie dokładności algorytmów dla różnego procentowego podziału datasetu na dane treningowe i testowe. Numeracja algorytmów na podstawie punktu 1 sprawozdania.

## 4.1 Fall Detection Data from China

Procent danych treningowych	Numer algorytmu				
	1	2	3	4	5
10%	0.559	0.161	0.296	0.551	0.29
20%	0.618	0.159	0.306	0.598	0.392
30%	0.619	0.18	0.302	0.61	0.302
40%	0.647	0.136	0.318	0.626	0.393
50%	0.657	0.135	0.299	0.644	0.387
60%	0.668	0.132	0.298	0.65	0.307
70%	0.695	0.139	0.306	0.65	0.391
80%	0.687	0.155	0.319	0.661	0.332
90%	0.692	0.146	0.329	0.668	0.348

Tabela 1: Porównanie dokładności algorytmu dla datasetu 1

## 4.2 Rain in Australia

Procent danych treningowych	Numer algorytmu				
	1	2	3	4	5
10%	1.0	0.938	0.78	0.839	0.976
20%	1.0	0.938	0.783	0.846	0.978
30%	1.0	0.944	0.784	0.848	0.967
40%	1.0	0.941	0.789	0.857	0.98
50%	1.0	0.943	0.8	0.86	0.981
60%	1.0	0.943	0.807	0.86	0.994
70%	1.0	0.945	0.81	0.866	0.977
80%	1.0	0.943	0.82	0.862	0.995
90%	1.0	0.942	0.816	0.857	0.991

Tabela 2: Porównanie dokładności algorytmu dla datasetu 2

## 4.3 Suicide Rates Overview 1985 to 2016

Procent danych treningowych	Numer algorytmu				
	1	2	3	4	5
10%	0.211	0.172	0.109	0.227	0.36
20%	0.254	0.202	0.137	0.246	0.367
30%	0.22	0.175	0.211	0.215	0.328
40%	0.231	0.183	0.183	0.22	0.403
50%	0.224	0.228	0.244	0.195	0.456
60%	0.215	0.187	0.264	0.212	0.538
70%	0.221	0.262	0.254	0.208	0.331
80%	0.198	0.229	0.302	0.219	0.538
90%	0.25	0.213	0.088	0.229	0.36

Tabela 3: Porównanie dokładności algorytmu dla datasetu 3

## 5 Wnioski

1. Wniosek1
2. Wniosek2