

Przedmiot: Sztuczna Inteligencja

TEMAT : Projekt końcowy

Jan Uthke Grupa : L6

# Spis treści

<b>Wstęp</b>	<b>3</b>
<b>1 Opisy wybranych metod</b>	<b>4</b>
1.1 Naive Bayes . . . . .	4
1.2 Decision Tree . . . . .	4
1.3 K-Nearest Neighbours . . . . .	5
<b>2 Kody wybranych metod</b>	<b>6</b>
2.1 Naive Bayes . . . . .	6
2.2 Decision Tree . . . . .	7
2.3 K-Nearest Neighbours . . . . .	8
<b>3 Analiza porównawcza</b>	<b>10</b>
3.1 Naive Bayes . . . . .	10
3.2 Decision Tree . . . . .	11
3.3 K-Nearest Neighbours . . . . .	12

# Wstęp

W niniejszym projekcie przedstawię wyniki mojej pracy nad zbiorem danych "Medical Cost Personal Datasets".

Link do repozytorium: [https://github.com/janonaj/projekt\\_SI](https://github.com/janonaj/projekt_SI)

Link do opracowania: <https://www.overleaf.com/read/nhztcpdznpcz>

# Rozdział 1

## Opisy wybranych metod

### 1.1 Naive Bayes

Model Naive Bayes to stara metoda klasyfikacji i predyktorów, która cieszy się renesansem ze względu na jego prostotę i stabilność. Problemy, z którymi model Naive Bayes są zazwyczaj stosowane, należą do dwóch szerokich kategorii: wyboru predyktora i klasyfikacji.

**Dobór predyktorów.** Są to aplikacje, w których wybierany jest podzbiór predyktorów z większego zestawu zmiennych. Większość metod klasyfikacji nie jest sprawna, gdy istnieje zbyt wiele predyktorów. Ponieważ w praktyce wiele predyktorów nie przyczynia się do klasyfikacji, to krok klasyfikacji wstępnej jest znalezienie podzbioru predyktorów, które są istotne.

**Klasyfikacja.** Są to aplikacje, w których używane są znane wartości jednej lub większej liczby zmiennych niezależnych (lub predyktorów) w celu oszacowania wartości jakościowej zmiennej przewidywanej.

Komponent Naive Bayes jest doskonałym narzędziem dla jednego z tych typów problemów, ale jest najbardziej przydatny w aplikacjach wymagających wyboru predyktorów, a następnie klasyfikacji. [1]

### 1.2 Decision Tree

Procedura drzew decyzyjnych tworzy model klasyfikacji oparty na drzewie. Klasyfikuje obserwacje w grupy lub przewiduje wartości zależnej (przewidywanej) zmiennej w oparciu o wartości niezależnych zmiennych (predyktorów). Ta procedura udostępnia narzędzia walidacyjne przeznaczone do analizy eksploracyjnej lub potwierdzającej.

Ta procedura może być używana na potrzeby:

**Segmentacja.** Identyfikacja osób, które mogą należeć do konkretnej grupy.

**Stratyfikacja.** Przypisywanie obserwacji do jednej z kilku kategorii, takich jak grupy wysokiego, średniego i niskiego ryzyka.

**Predykcja.** Tworzenie reguł i używanie ich w celu przewidywania zdarzeń w przyszłości, takich jak prawdopodobieństwo tego, że ktoś nie będzie spłacał pożyczki, albo potencjalna wartość samochodu lub domu przy odsprzedaży.

**Redukcja danych i filtrowanie zmiennych.** Wybór użytecznego podzbioru predyktorów z dużego zestawu zmiennych w celu opracowania formalnego modelu parametrycznego.

**Identyfikacja interakcji.** Identyfikacja zależności, które dotyczą tylko konkretnych grup, a następnie określenie ich w formalnym modelu parametrycznym.

**Scalanie kategorii i dyskretyzacja zmiennych ciągłych.** Ponowne kodowanie kategorii predyktorów grup i zmiennych ciągłych z minimalnymi stratami informacji.[2]

## 1.3 K-Nearest Neighbours

Analiza najbliższego sąsiedztwa jest metodą klasyfikacji obserwacji na podstawie ich podobieństwa do innych obserwacji. Zostało to opracowane w nauczaniu maszynowym jako sposób rozpoznawania wzorców danych bez konieczności zapewnienia dokładnej zgodności z jakimikolwiek zapamiętanymi wzorcami lub obserwacjami. Podobne obserwacje znajdują się blisko siebie, a niepodobne — daleko. Zatem odległość między dwoma obserwacjami stanowi miarę ich niepodobieństwa.

Obserwacje znajdujące się blisko siebie nazywają się „sąsiedztwem”. Podczas prezentacji nowej (wstrzymanej) obserwacji, obliczana jest odległość od każdej obserwacji modelu. Zostaje określona klasyfikacja najbardziej podobnych obserwacji najbliższego sąsiedztwa, a nowa obserwacja zostaje umieszczona w kategorii, która zawiera największą liczbę obserwacji najbliższego sąsiedztwa.

Można określić liczbę najbliższych elementów sąsiednich do analizowania; ta wartość to  $k$ . Na zdjęciach przedstawiono, w jaki sposób nowa obserwacja zostanie sklasyfikowana przy użyciu dwóch różnych wartości  $k$ . Gdy  $k = 5$ , nowa obserwacja jest umieszczana w kategorii 1, ponieważ większość obserwacji najbliższego sąsiedztwa należy do kategorii 1. Jednak gdy  $k = 9$ , nowa obserwacja jest umieszczana w kategorii 0, ponieważ większość obserwacji najbliższego sąsiedztwa należy do kategorii 0.

Analiza najbliższego sąsiedztwa może być również użyta do obliczania docelowych wartości ilościowych. W tej sytuacji do uzyskania przewidywanej wartości dla nowej obserwacji stosowana jest docelowa wartość średniej lub mediany najbliższych sąsiadów.[3]

## Rozdział 2

# Kody wybranych metod

### 2.1 Naive Bayes

```
1 library(naivebayes)
2
3 dane = read.csv("C://Users//janut//Desktop//SI//insurance.csv",
4 header = TRUE, sep = ",")
5
6
7
8 Dokladnosc = function(){
9
10     idx=sample(2,nrow(dane),replace=T,prob=c(0.8,0.2))
11     train = dane[idx==1,]
12     test = dane[idx==2,]
13
14     model=naive_bayes(as.character(smoker) ~ .,
15 data=train, usekernel=T)
16     plot(model)
17
18     #PREDYKCJA
19     p = predict(model, test)
20     cm = table(p, test$smoker)
21
22     #print(cm)
23     pcf = cm/sum(cm)
24
25     #DOKŁADNOŚĆ
26     return (sum(diag(cm))/sum(cm))
27 }
28 results = replicate(100, Dokladnosc())
29
```

```

30 #Policzenie odchylenia standardowej
31 sr_dokladnosc = mean(results)
32 odchylenie = sd(results)
33
34 #Wyświetl wynik
35 cat("Jakosc klasyfikatora:", sr_dokladnosc, "\n")
36 cat("Odchylenie standardowe:", odchylenie, "\n")

```

Rys. 2.1: NaiveBayes

## 2.2 Decision Tree

```

1 void SortowanieBabelkowe( int tab[], int size )
2 library(caret)
3 library(rpart)
4
5 #wczytywanie danych
6 data = read.csv("C://Users//janut//Desktop//SI//insurance.csv",
7 header = TRUE, sep = ",")
8
9 #konwersja kolumny "smoker" na faktory
10 data$smoker = as.factor(data$smoker)
11
12 #powtorzenie procesu trzykrotnie
13 accuracies = c()
14 for(i in 1:3){
15     #podział danych na zbiór treningowy i testowy
16     trainIndex = sample(1:nrow(data), 0.7 * nrow(data))
17     trainData = data[trainIndex, ]
18     testData = data[-trainIndex, ]
19
20     #wytrenowanie modelu drzewa decyzyjnego
21     library(rpart)
22     fit = rpart(smoker ~., data = trainData, method = "class")
23
24     #ocena jakości modelu
25     pred = predict(fit, newdata = testData, type = "class")
26     confMatrix = confusionMatrix(pred, testData$smoker)
27
28     #zapisanie dokładności
29     accuracies[i] = confMatrix$overall[1]
30 }
31
32 #obliczanie odchylenia standardowego i średniej dokładności

```

```

33 meanAccuracy = mean(accuracies)
34 sdAccuracy = sd(accuracies)
35
36 print(paste("Średnia dokładność", meanAccuracy))
37 print(paste("Odchylenie standardowe", sdAccuracy))

```

Rys. 2.2: DT

## 2.3 K-Nearest Neighbours

```

1  # Wczytanie biblioteki knn
2  library(class)
3
4  # Wczytywanie danych z pliku
5  dane = read.csv("C://Users//janut//Desktop//SI//insurance.csv",
6  header = TRUE, sep = ",")
7  dane=dane[,-6] #usuwa region
8
9
10 # Stwórz mapowanie wartości na liczby
11 value_mapping <- c("male" = 1, "female" = 2)
12
13 # Zastąp dane charakteru numerycznymi danymi
14 dane$sex <- value_mapping[as.character(dane$sex)]
15
16
17 dane$smoker = as.factor(dane$smoker)
18
19 # Funkcja do wyliczenia jakości klasyfikacji
20 jakosc_klasyfikacji <- function(training, test, k) {
21   print(training[, -5])
22   model = knn(training[, -5], test[, -5], training[, 5], k = k)
23   confusion_matrix <- table(test[, 5], model)
24   jakosc = sum(diag(confusion_matrix)) / sum(confusion_matrix)
25   #print(confusion_matrix)
26   #print(jakosc)
27   return(jakosc)
28 }
29
30 # Powtórzenie obliczeń kilkukrotnie
31 jakosc_klasyfikacji_xn <- replicate(100, {
32   idx=sample(2,nrow(dane),replace=T,prob=c(0.8,0.2))
33   training_data=dane[idx==1,]
34   test_data=dane[idx==2,]

```



```

35     jakosc_klasyfikacji(training_data, test_data, k = 8)
36 })
37
38 # Policzenie odchylenia standardowej
39 sr_dokladnosc = mean(jakosc_klasyfikacji_xn)
40 odchylenie = sd(jakosc_klasyfikacji_xn)
41
42 # Wyświetl wynik
43 cat("Jakosc klasyfikatora:", sr_dokladnosc, "\n")
44 cat("Odchylenie standardowe:", odchylenie, "\n")

```

Rys. 2.3: KNN

## Rozdział 3

# Analiza porównawcza

### 3.1 Naive Bayes

Confusion matrix dla metody Naive Bayes.

#### 1. Wyniki dla stosunku 0,8 : 0,2

p	no	yes
no	211	12
yes	4	34

Jakosc klasyfikatora: 0.9124

Odchylenie standardowe: 0.0181304

#### 2. Wyniki dla stosunku 0,7 : 0,3

p	no	yes
no	303	36
yes	11	69

Jakosc klasyfikatora: 0.9111012

Odchylenie standardowe: 0.01409661

#### 3. Wyniki dla stosunku 0,6 : 0,4

p	no	yes
no	381	22
yes	27	80

Jakosc klasyfikatora: 0.9105262

Odchylenie standardowe: 0.0119198

## 3.2 Decision Tree

Confusion matrix dla metody DT.

### 1. Wyniki dla stosunku 0,8 : 0,2

p	no	yes
no	202	2
yes	4	60

Średnia dokładność 0.972636815920398

Odchylenie standardowe 0.0155348208915383

### 2. Wyniki dla stosunku 0,7 : 0,3

p	no	yes
no	313	12
yes	6	71

Średnia dokładność 0.962686567164179

Odchylenie standardowe 0.0108430321978624

### 3. Wyniki dla stosunku 0,6 : 0,4

p	no	yes
no	404	8
yes	16	108

Średnia dokładność 0.953980099502488

Odchylenie standardowe 0.00569972101362671

### 3.3 K-Nearest Neighbours

Confusion matrix dla metody KNN po usunięciu cechy nr 6 - "region".

#### 1. Wyniki dla stosunku 0,8 : 0,2

p	no	yes
no	202	12
yes	9	50

Jakość klasyfikatora: 0.9232172

Odchylenie standardowe: 0.01495778

#### 2. Wyniki dla stosunku 0,7 : 0,3

p	no	yes
no	284	10
yes	13	78

Jakość klasyfikatora: 0.9220333

Odchylenie standardowe: 0.01095005

#### 3. Wyniki dla stosunku 0,6 : 0,4

p	no	yes
no	392	25
yes	23	93

Jakość klasyfikatora: 0.9219228

Odchylenie standardowe: 0.01002492

# Listings

2.1	NaiveBayes . . . . .	6
2.2	DT . . . . .	7
2.3	KNN . . . . .	8

# Bibliografia

- [1] I. B. M. Corporation. Naive bayes (naivebayes). *IBM SPSS Statistics dokumentacja*, 2022-09-13. <https://www.ibm.com/docs/pl/spss-statistics/saas?topic=edition-naive-bayes>.
- [2] I. B. M. Corporation. Tworzenie drzew decyzyjnych. *IBM SPSS Statistics dokumentacja*, 2022-09-13. <https://www.ibm.com/docs/pl/spss-statistics/saas?topic=trees-creating-decision>.
- [3] I. B. M. Corporation. Węzeł knn). *IBM SPSS Statistics dokumentacja*, 2022-12-01. <https://www.ibm.com/docs/pl/spss-modeler/18.4.0?topic=models-knn-node>.