Przedmiot: Sztuczna Inteligencja

TEMAT : Projekt końcowy

Jan Uthke Grupa : L6

# Spis treści

# Wstęp

W niniejszym projekcie przedstadstawione są wyniki mojej pracy nad zbiorem danych "Medical Cost Personal Datasets". W efekcie przeprowadzona została analiza porównawcza wyników klasyfikacji uzyskanych z trzech wybranych przeze mnie metod: Naive Bayes, Decision Tree, oraz K-Nearest Neighbours.

Link do repozytorium: https://github.com/janonaj/projekt_SI

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

Źródło repozytorium: https://www.kaggle.com/datasets/mirichoi0218/insurance

# 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 predyletów, 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

```
library(naivebayes)

dane = read.csv("C://Users//janut//Desktop//SI//insurance.csv",
header = TRUE, sep = ",")



Dokladnosc = function(){

  idx=sample(2,nrow(dane),replace=T,prob=c(0.8,0.2))
  train = dane[idx==1,]
  test = dane[idx==2,]

  model=naive_bayes(as.character(smoker) ~ .,
  data=train, usekernel=T)
  plot(model)

  #PREDYKCJA
  p = predict(model, test)
  cm = table(p, test$smoker)

  #print(cm)
  pcf = cm/sum(cm)

  #DOKŁADNOŚĆ
  return (sum(diag(cm))/sum(cm))
}
results = replicate(100, Dokladnosc())

```

```
#Policzenie odchylenia standardowej
sr_dokladnosc = mean(results)
odchylenie = sd(results)

#Wyświetl wynik
cat("Jakosc klasyfikatora:", sr_dokladnosc, "\n")
cat("Odchylenie standardowe:", odchylenie, "\n")
```

Rys. 2.1: NaiveBayes

## 2.2 Decision Tree

```
void SortowanieBabelkowe( int tab[], int size )
library(caret)
library(rpart)

#wczytywanie danych
data = read.csv("C://Users//janut//Desktop//SI//insurance.csv",
header = TRUE, sep = ",")

#konwersja kolumny "smoker" na faktory
data$smoker = as.factor(data$smoker)

#powtorzenie procesu trzykrotnie
accuracies = c()
for(i in 1:3){
  #podział danych na zbiór treningowy i testowy
  trainIndex = sample(1:nrow(data), 0.7 * nrow(data))
  trainData = data[trainIndex, ]
  testData = data[-trainIndex, ]

  #wytrenowanie modelu drzewa decyzyjnego
  library(rpart)
  fit = rpart(smoker ~., data = trainData, method = "class")

  #ocena jakości modelu
  pred = predict(fit, newdata = testData, type = "class")
  confMatrix = confusionMatrix(pred,testData$smoker)

  #zapisanie dokładności
  accuracies[i] = confMatrix$overall[1]
}

#obliczanie odchylenia standardowego i średniej dokładności
```

```
meanAccuracy = mean(accuracies)
sdAccuracy = sd(accuracies)

print(paste("Średnia dokładność", meanAccuracy))
print(paste("Odchylenie standardowe", sdAccuracy))
```

Rys. 2.2: DT

## 2.3 K-Nearest Neighbours

```
# Wczytanie biblioteki knn
library(class)

# Wczytywanie danych z pliku
dane = read.csv("C://Users//janut//Desktop//SI//insurance.csv",
header = TRUE, sep = ",")
dane=dane[,-6] #usuwa region


# Stwórz mapowanie wartości na liczby
value_mapping <- c("male" = 1, "female" = 2)

# Zastąp dane charakteru numerycznymi danymi
dane$sex <- value_mapping[as.character(dane$sex)]


dane$smoker = as.factor(dane$smoker)

# Funkcja do wyliczenia jakości klasyfikacji
jakosc_klasyfikacji <- function(training, test, k) {
  print(training[, -5])
  model = knn(training[, -5], test[, -5], training[, 5], k = k)
  confusion_matrix <- table(test[, 5], model)
  jakosc = sum(diag(confusion_matrix)) / sum(confusion_matrix)
  #print(confusion_matrix)
  #print(jakosc)
  return(jakosc)
}

# Powtórzenie obliczeń kilkukrotnie
jakosc_klasyfikacji_xn <- replicate(100, {
  idx=sample(2,nrow(dane),replace=T,prob=c(0.8,0.2))
  training_data=dane[idx==1,]
  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

# 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.