Sprawozdanie 01 - Bayes i KNN

Szymon Kolodziejski

March 2024

1 Klasyfikator

- Naiwny Bayes (z j.ang. Naive Bayes) - zakłada niezależność między cechami i oblicza prawodopodobieństwo przynależności danych wejściowych do określonej klasy na podstawie następującego wzoru:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{1}$$

  gdzie

  - $P(c|x)$ - prawdopodobieństwo wystąpienia klasy c, jeśli wystąpi klasa x
  - $P(c)$ - prawdopodobieństwo wystąpienia klasy c
  - $P(x|c)$ - prawdopodobieństwo wystąpienia klasy x, jeśli wystąpi klasa c
  - $P(c)$ - prawdopodobieństwo wystąpienia klasy x

- K-najbliższych sąsiadów (z j.ang. k-nearest neighbor (KNN)) - klasyfikator przypisujący najbliższą oraz najliczniejszą etykietę do nieoznaczonych obserwacji. Na podstawie dystansu (domyślnie jest to odległość euklidesowa) między punktami (etykietowanym oraz przewidywanym), które wyznacza się następująco:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{2}$$

  gdzie x i y są podmiotami, które porównujemy z n-tą cechą.

2 Pakiety

```
> install.packages(c("kernlab", "tidyverse", "e1071",
                     "caTools", "caret", "class"))
> library(kernlab)
> library(tidyverse)
```

```
> library(e1071)
> library(caTools)
> library(caret)
> library(class)
> library(caret)
```

3 Bazy danych

- iris

  ```
  > data("iris")
  ```

- spam

  ```
  > data(spam)
  ```

4 Wstępne przetwarzanie danych

- iris

  ```
  > rows <- sample.int(nrow(iris), size = round(nrow(iris)/3), replace = F)
  > iris.train <- iris[-rows,]
  > iris.test <- iris[rows,]
  ```

- spam

  ```
  > rows <- sample.int(nrow(spam), size = round(nrow(spam)/3), replace = F)
  > spam.train <- spam[-rows,]
  > spam.test <- spam[rows,]
  ```

5 Bayes

- iris

  ```
  > iris.NB <- naiveBayes(x = subset(iris.train, select = -c(Species)),
  +                       y = iris.train['Species'])
  > iris.NB.pred <- predict(iris.NB, newdata = iris.test)
  > confusion_matrix <- table(iris.NB.pred, iris.test$Species)
  > accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
  > print(confusion_matrix)
  iris.NB.pred setosa versicolor virginica
    setosa         19          0         0
    versicolor      0         15         0
    virginica       0          1        15
  > cat("Accuracy:", round(accuracy * 100, 2), "%\n")
  Accuracy: 98 %
  ```

- spam

```
> spam.NB <- naiveBayes(x = subset(spam.train, select = -c(type)),
+                       y = spam.train['type'])
> spam.NB.pred <- predict(spam.NB, newdata = spam.test)
> spam.confusion_matrix <- table(spam.NB.pred, spam.test$type)
> spam.accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> print(spam.confusion_matrix)
spam.NB.pred nonspam spam
     nonspam     488   28
     spam        421  597
> cat("Accuracy:", round(spam.accuracy * 100, 2), "%\n")
Accuracy: 98 %
```

6 KNN

- iris

```
> iris.KNN.tC <- trainControl(method  = "cv",
+                             number  = 5)
> iris.KNN.fit <- train(Species ~ .,
+                       method     = "knn",
+                       tuneGrid   = expand.grid(k = 1:10),
+                       trControl  = iris.KNN.tC,
+                       metric     = "Accuracy",
+                       data       = iris)
> iris.KNN.fit
k-Nearest Neighbors

150 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 120, 120, 120, 120, 120
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.9600000  0.94
   2  0.9533333  0.93
   3  0.9600000  0.94
   4  0.9666667  0.95
   5  0.9666667  0.95
   6  0.9733333  0.96
   7  0.9733333  0.96
```

```
   8  0.9800000  0.97
   9  0.9800000  0.97
  10  0.9733333  0.96

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
> cat("Accuracy:", mean(iris.KNN.fit[["results"]][["Accuracy"]]))
Accuracy: 0.9686667
```

- spam

```
> spam.KNN.tC <- trainControl(method  = "cv",
+                             number  = 5)
> spam.KNN.fit <- train(type ~ .,
+                       method     = "knn",
+                       tuneGrid   = expand.grid(k = 1:10),
+                       trControl  = iris.KNN.tC,
+                       metric     = "Accuracy",
+                       data       = spam)
> spam.KNN.fit
k-Nearest Neighbors

4601 samples
  57 predictor
   2 classes: 'nonspam', 'spam'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3680, 3681, 3681, 3681, 3681
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.8161240  0.6159760
   2  0.7852637  0.5528897
   3  0.8006958  0.5830824
   4  0.7989591  0.5783041
   5  0.8070007  0.5952230
   6  0.7933062  0.5654954
   7  0.7915701  0.5605889
   8  0.7909172  0.5589357
   9  0.7846169  0.5455314
  10  0.7830923  0.5432034

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 1.
> cat("Accuracy:", mean(spam.KNN.fit[["results"]][["Accuracy"]]))
```

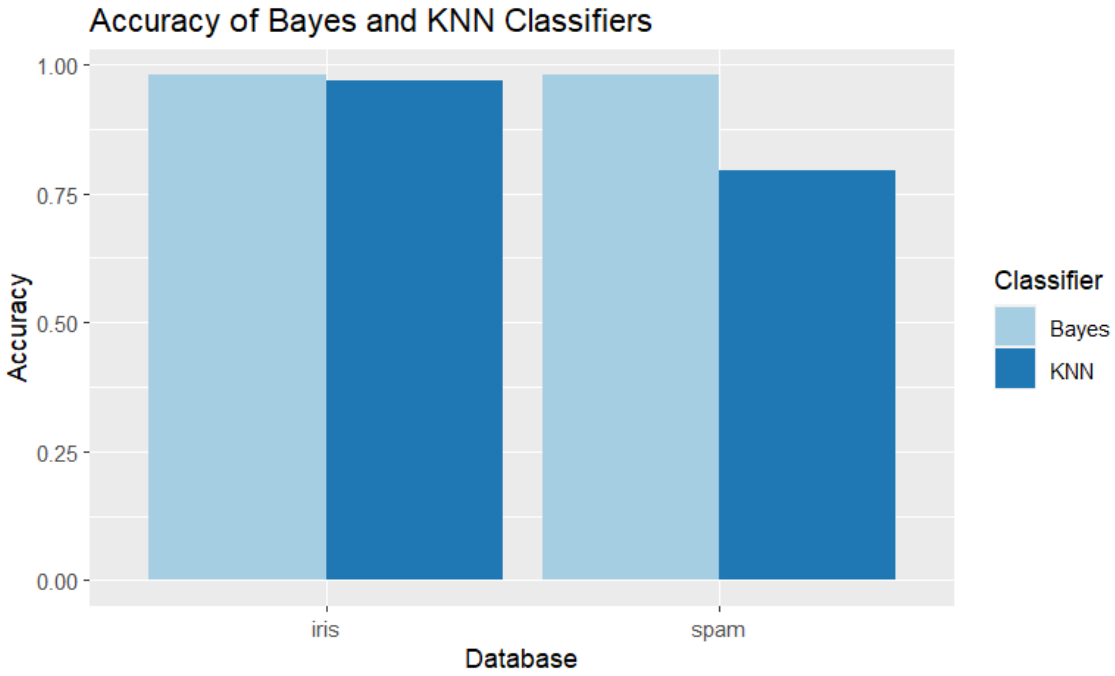
```
Accuracy: 0.7951546
```

7 Podsumowanie

	Klasyfikator	
baza danych	Bayes	KNN
iris	0.98	0.968666
spam	0.98	0.795154

Tabela 1: ACC

```
> library(ggplot2)
> data <- data.frame(
+   Database = c("iris", "spam", "iris", "spam"),
+   Classifier = c("Bayes", "Bayes", "KNN", "KNN"),
+   Accuracy = c(0.98, 0.98, 0.968666, 0.795154)
+ )
> plot <- ggplot(data, aes(x = Database, y = Accuracy, fill = Classifier)) +
+   geom_bar(stat = "identity", position = "dodge") +
+   labs(title = "Accuracy of Bayes and KNN Classifiers",
+        x = "Database",
+        y = "Accuracy",
+        fill = "Classifier") +
+   scale_fill_brewer(palette = "Paired") +
+   theme_minimal() +
+   theme(plot.title = element_text(hjust = 0.5))
> print(plot)
```



Rysunek 1: ACC bar plot

Na wykresie 1 można dostrzec przewagę naiwnego klasyfikatora Bayes'a (z j. ang. naive Bayes classifier) nad k najbliższych sąsiadów (z j.ang. k-nearest neighbors (KNN)) w przypadku miej zbalansowanej bazy danych "spam". Dokładność klasyfikatora Bayes'a dla baz danych iris oraz spam jest równa 0.98 przy błędzie wynoszącym 0.02. W opozycji do klasyfikatora KNN, gdzie dokładność dla zbioru iris wynosi około 0.97, przy błędzie wynoszącym 0.03, z kolei dla spam jest znacznie niższa czyli około 0.8, przy błędzie opiewającym na około 0.2.