Министерство образования и науки РФ
Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1

«Наивный Байесовский классификатор»
по дисциплине «Машинное обучение»

Выполнила:
студентка гр. 3540201/20301 _____ Климова О. А.
подпись, дата

Проверил:
д.т.н., проф. _____ Уткин Л. В.
подпись, дата

Санкт-Петербург
2022

Содержание

Постановка задачи

В рамках данной работы необходимо:

1. Исследовать, как объем обучающей выборки и количество тестовых данных, влияет на точность классификации или на вероятность ошибочной классификации в примере крестики-нолики и примере о спаме e-mail сообщений.

2. Сгенерировать 100 точек с двумя признаками X1 и X2 в соответствии с нормальным распределением так, что первые 50 точек (class -1) имеют параметры: мат. ожидание X1 равно 10, мат. ожидание X2 равно 14, среднеквадратические отклонения для обеих переменных равны 4. Вторые 50 точек (class +1) имеют параметры: мат. ожидание X1 равно 20, мат. ожидание X2 равно 18, среднеквадратические отклонения для обеих переменных равны 3. Построить соответствующие диаграммы, иллюстрирующие данные. Построить байесовский классификатор и оценить качество классификации.

3. Разработать байесовский классификатор для данных Титаник (Titanic dataset) - https://www.kaggle.com/c/titanic.

1 Исследование зависимости точности классификатора от размера выборки

1.1 «Крестики-нолики»

Рассмотрим выборку из 958 элементов, каждый из которых для определенного расположения в игре «Крестики-нолики» классифицирует его как выигрышное (positive) или проигрышное (negative) для x:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
623	b	b	b	x	x	x	b	o	o	positive
624	b	b	b	o	o	b	x	x	x	positive
625	b	b	b	o	b	o	x	x	x	positive
626	b	b	b	b	o	o	x	x	x	positive
627	x	x	o	x	x	o	o	b	o	negative
628	x	x	o	x	x	o	b	o	o	negative
629	x	x	o	x	x	b	o	o	o	negative

Отобразим зависимость точности классификатора от размера тестовой и обучающей выборки из датасета «Крестики-нолики».

Для оценки точности используем метрику accuracy, где TP и TN - верно классифицируемые объекты:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

При обучающей выборке 90%, а тренировочной 10%

```
A_predicted negative positive
   negative       13        7
   positive       18       58
> accuracy
[1] 0.7395833
```

При обучающей выборке 80%, а тренировочной 20%

```
A_predicted negative positive
   negative       22       20
   positive       40      110
> accuracy
[1] 0.6875
```

При обучающей выборке 70%, а тренировочной 30%

```
A_predicted negative positive
   negative       34       38
   positive       53      163
> accuracy
[1] 0.6840278
```

При обучающей выборке 60%, а тренировочной 40%

```
A_predicted negative positive
   negative       50       43
   positive       74      217
> accuracy
[1] 0.6953125
```

При обучающей выборке 50%, а тренировочной 50%

```
A_predicted negative positive
   negative       66       51
   positive       91      271
> accuracy
[1] 0.7035491
```

При обучающей выборке 40%, а тренировочной 60%

```
A_predicted negative positive
   negative       80       63
   positive      115      317
> accuracy
[1] 0.6904348
```

При обучающей выборке 30%, а тренировочной 70%

```
A_predicted negative positive
   negative       96       73
   positive      132      370
> accuracy
[1] 0.6944858
```

При обучающей выборке 20%, а тренировочной 80%

```
A_predicted negative positive
   negative      119      104
   positive      139      405
> accuracy
[1] 0.6831812
```

При обучающей выборке 10%, а тренировочной 90%

```
A_predicted negative positive
   negative      133      104
   positive      163      463
> accuracy
[1] 0.6906141
```

1.2 Классификация спама

Рассмотрим выборку из 4601 элемента, каждый из которых для набора признаков текстового сообщения классифицирует его как спам или не спам:

charExclamation	charDollar	charHash	capitalAve	capitalLong	capitalTotal	type
0.778	0.000	0.000	3.756	61	278	spam
0.372	0.180	0.048	5.114	101	1028	spam
0.276	0.184	0.010	9.821	485	2259	spam
0.137	0.000	0.000	3.537	40	191	spam
0.135	0.000	0.000	3.537	40	191	spam
0.000	0.000	0.000	3.000	15	54	spam
0.164	0.054	0.000	1.671	4	112	spam

Отобразим зависимость точности классификатора от размера тестовой и обучающей выборки из датасета «Спам».

Для оценки точности используем метрику accuracy, где TP и TN - верно классифицируемые объекты:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

При обучающей выборке 90%, а тренировочной 10%

```
          nonspam spam
  nonspam     147    7
  spam        134  172
> accuracy2
[1] 0.6934783
```

При обучающей выборке 80%, а тренировочной 20%

```
          nonspam spam
  nonspam     303   21
  spam        249  347
> accuracy2
[1] 0.7065217
```

При обучающей выборке 70%, а тренировочной 30%

```
          nonspam spam
  nonspam     440   25
  spam        375  540
> accuracy2
[1] 0.7101449
```

При обучающей выборке 60%, а тренировочной 40%

```
          nonspam spam
  nonspam     661   44
  spam        454  681
> accuracy2
[1] 0.7293478
```

При обучающей выборке 50%, а тренировочной 50%

```
          nonspam spam
  nonspam     798   39
  spam        570  893
> accuracy2
[1] 0.7352174
```

При обучающей выборке 40%, а тренировочной 60%

```
          nonspam spam
  nonspam    1007   57
  spam        692 1004
> accuracy2
[1] 0.7286232
```

При обучающей выборке 30%, а тренировочной 70%

```
          nonspam spam
  nonspam    1107   60
  spam        853 1200
> accuracy2
[1] 0.7164596
```

При обучающей выборке 20%, а тренировочной 80%

```
          nonspam spam
  nonspam    1205   91
  spam       1030 1354
> accuracy2
[1] 0.6953804
```

При обучающей выборке 10%, а тренировочной 90%

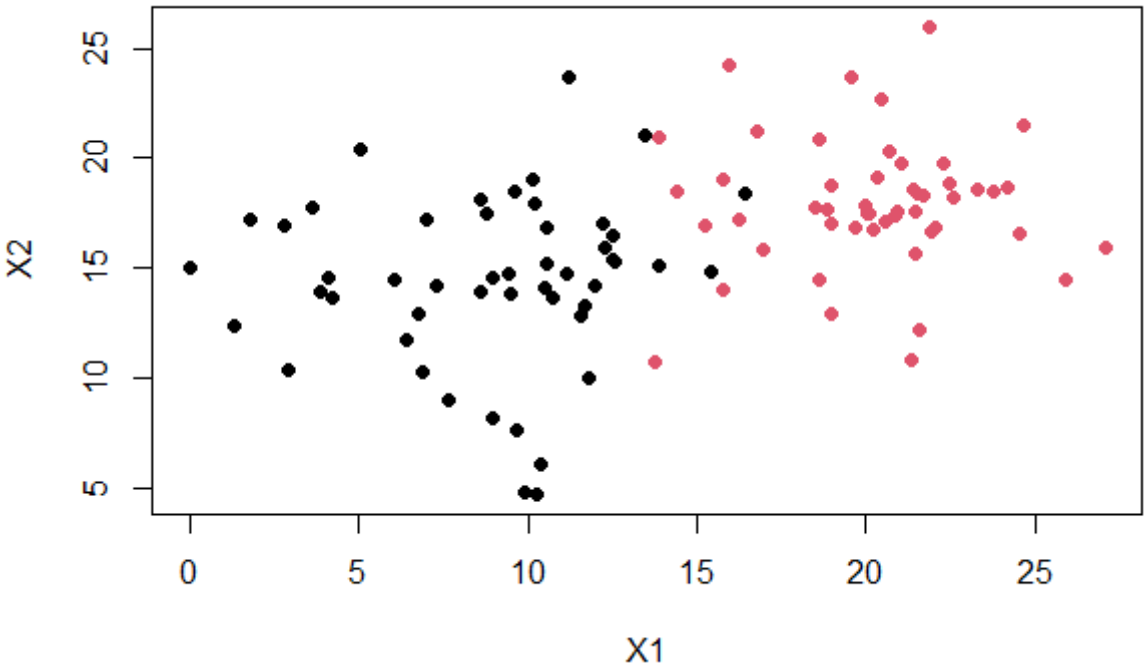
```
          nonspam spam
  nonspam    1258   81
  spam       1246 1555
> accuracy2
[1] 0.6794686
```

1.3 Выводы

Можно сделать вывод, что для обоих датасетов «Крестики-нолики» и «Спам» увеличение числа примеров в обучающей выборке не приводит к улучшению качества работы Наивного Байесовского классификатора.

2 Генерация точек

Сгенерируем два кластера точек в соответствии с заданием. Результат будет иметь следующий вид:



Если для обучения использовать 80% точек (80 штук), а для теста 20% (20 штук), то на тестовой выборке классификатор выдает точность accuracy = 0.95:

```
T_predicted -1  1
         -1  0  1
         1   0 19
> accuracy3
[1] 0.95
```

Если для обучения использовать 70% точек (70 штук), а для теста 30% (30 штук), то на тестовой выборке классификатор выдает точность accuracy = 0.93:

```
T_predicted -1  1
         -1  0  2
         1   0 28
> accuracy3
[1] 0.9333333
```

Если для обучения использовать 60% точек (60 штук), а для теста 40% (40 штук), то на тестовой выборке классификатор выдает точность accuracy = 0.75:

```
T_predicted -1  1
         -1  0 10
         1   0 30
> accuracy3
[1] 0.75
```

Можно видеть, что байесовский классификатор выдает точность на данном датасете выше, чем на предыдущих примерах.

3 Титаник

Для датасета «Титаник» была загружена обучающая выборка, включающая 891 элемент:

	PassengerId	Survived	Pclass	Name	Sex	Age
1	1	0	3	Braund, Mr. Owen Harris	male	22.00
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.00
3	3	1	3	Heikkinen, Miss. Laina	female	26.00
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00
5	5	0	3	Allen, Mr. William Henry	male	35.00
6	6	0	3	Moran, Mr. James	male	*NA*
7	7	0	1	McCarthy, Mr. Timothy J	male	54.00
8	8	0	3	Palsson, Master. Gosta Leonard	male	2.00
9	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.00

А также тестовая выборка, включающая 418 элементов:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch
1	892	3	Kelly, Mr. James	male	34.5	0	0
2	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0
3	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0
4	895	3	Wirz, Mr. Albert	male	27.0	0	0
5	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1
6	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0
7	898	3	Connolly, Miss. Kate	female	30.0	0	0
8	899	2	Caldwell, Mr. Albert Francis	male	26.0	1	1

Далее на основе обучающих данных был построен Байесовский классификатор, использующий такие признаки, как класс кают, имя, пол, возраст, число братьев-сестер/муж-жена на борту; число родителей/детей на борту; номер билета; стоимость билета; каюта.

Затем на тестовой выборке было проведено тестирование и получена точность accuracy = 0.75:

```
Titanic_predicted   0   1
                0 232  69
                1  34  83
> accuracy4
[1] 0.7535885
```

Приложение 1. Код примеров, используемых для исследования точности классификаторов

```
##КРЕСТИКИ-НОЛИКИ
#install.packages("e1071")
library(e1071)
# импортируем данные в R
A_raw<-read.table("C:/Users/Unicorn/Desktop/МашинноеОбучение/Лабы/Tic_tac_toe.txt",
sep = ",", stringsAsFactors = TRUE)
# число строк в базе
n <- dim(A_raw)[1]
# Устанавливаем базу генерации случайных чисел и рандомизируем выборку
set.seed(12345)
A_rand <- A_raw[ order(runif(n)), ]
# разделим данные на обучающие и тестирующие (80% для обучения)
nt <- as.integer(n*0.8)
A_train <- A_rand[1:nt, ]
A_test <- A_rand[(nt+1):n, ]
# Используем Наивный Байесовский классификатор из пакета e1071
A_classifier <- naiveBayes(V10 ~ ., data = A_train)
# Теперь оценим полученную модель:
A_predicted <- predict(A_classifier, A_test)
# Используем table для сравнения прогнозируемых значений с тем, что есть
tab1 = table(A_predicted, A_test$V10)
# Вычислим точность
accuracy = (tab1[1,1] + tab1[2,2]) / (tab1[1,1] + tab1[2,2] + tab1[1,2] + tab1[2,1])
# Выводим таблицу с числом классифицированных элементов и точность
tab1
accuracy

##СПАМ
library(kernlab)
library(e1071)
data(spam)
## Случайным образом выбираем 10% сообщений для тестирования,
idx <- sample(1:dim(spam)[1], 4601*0.9);
spamtrain <- spam[-idx, ];
spamtest <- spam[idx, ];
## Обучаем и оцениваем классификатор
model <- naiveBayes(type ~ ., data = spamtrain);
tab2 = table(predict(model, spamtest), spamtest$type)
accuracy2 = (tab2[1,1] + tab2[2,2]) / (tab2[1,1] + tab2[2,2] + tab2[1,2] + tab2[2,1])
# Выводим таблицу с числом классифицированных элементов и точность
tab2
accuracy2
```

Приложение 2. Код, используемый для генерации точек и обучения классификатора

```
##ГЕНЕРАЦИЯ ТОЧЕК
library(e1071)
n1 <- rnorm(50, 10, 4)
n2 <- rnorm(50, 20, 3)
n3 <- rnorm(50, 14, 4)
n4 <- rnorm(50, 18, 3)

X1 <- c(n1,n2)
X2 <- c(n3,n4)
C <- rep(c("-1" , "1") , each = 50)
T_rand <- data.frame(X1, X2, C, stringsAsFactors = TRUE)
plot(X1, X2, col = rep(1:2, each = 50), pch = 19)

# Разделим данные на обучающие и тестирующие
n = 100
#70% обучающая
nt <- as.integer(n*0.7)
T_train <- T_rand[1:nt, ]
T_test <- T_rand[(nt+1):n, ]

##Обучение
T_classifier <- naiveBayes(C ~ ., data = T_train)
# Оценка полученной модели:
T_predicted <- predict(T_classifier, T_test)
# Сравним прогнозируемые значения с тем, что есть
tab3 = table(T_predicted, T_test$C)
accuracy3 = (tab3[1,1] + tab3[2,2]) / (tab3[1,1] + tab3[2,2] + tab3[1,2] + tab3[2,1])
# Выводим таблицу с числом классифицированных элементов и точность
tab3
accuracy3
```

Приложение 3. Код, используемый для построения классификатора на основе датасета «Титаник»

```
##ТИТАНИК
library(e1071)
#Загрузка обучающей выборки
Titanic_train <- read.csv("Titanic_train.csv", header = TRUE, sep = ",", dec = ".",
                          stringsAsFactors = FALSE)
#Загрузка тестовой выборки
Titanic_test <- read.csv("Titanic_test.csv", header = TRUE, sep = ",", dec = ".",
                         stringsAsFactors = FALSE)

#Обучение
Titanic_classifier <- naiveBayes(Survived ~ ., data = Titanic_train)
# Оценка полученной модели:
Titanic_predicted <- predict(Titanic_classifier, Titanic_test)

# Сравним полученные результаты с тестовыми
Gen_sub = read.csv("gender_submission.csv", header = TRUE, sep = ",", dec = ".",
                   stringsAsFactors = FALSE)
tab4 = table(Titanic_predicted, Gen_sub$Survived)
accuracy4 = (tab4[1,1] + tab4[2,2]) / (tab4[1,1] + tab4[2,2] + tab4[1,2] + tab4[2,1])
# Выводим таблицу с числом классифицированных элементов и точность
tab4
accuracy4
```