САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. ПЕТРА ВЕЛИКОГО

Институт прикладной математики и механики

Высшая школа прикладной математики и вычислительной физики

Отчёт

по лабораторной работе №2 «Наивный Байесовский классификатор»
по дисциплине «Системы искусственного интеллекта»

Студентка гр. 3630201/70101 _____ О. В. Саксина

Преподаватель _____ Л. В. Уткин

Санкт-Петербург,
2021г.

Содержание

1 Задание 1

1.1 Постановка задачи

Исследуйте, 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.

1.2 Реализация

На рис. 1 и 2 представлены графики зависимости точности классификации от размера обучающей и тестовой выборки для данных крестики-нолики. При увеличении размера обучающей выборки, точность классификации на тестовых данных сначала повышается, так как новые данные позволяют модели лучше обучиться, но при достижении определённого уровня точность снижается из-за переобучения модели. При увеличении размера тестовой выборки точность также снижается, это может быть связано с характеристикой данных, попавших в тестовую выборку: при её малом размере вероятность того,что попадутся примеры с выбросами или какими-то нетипичными параметрами, ниже.

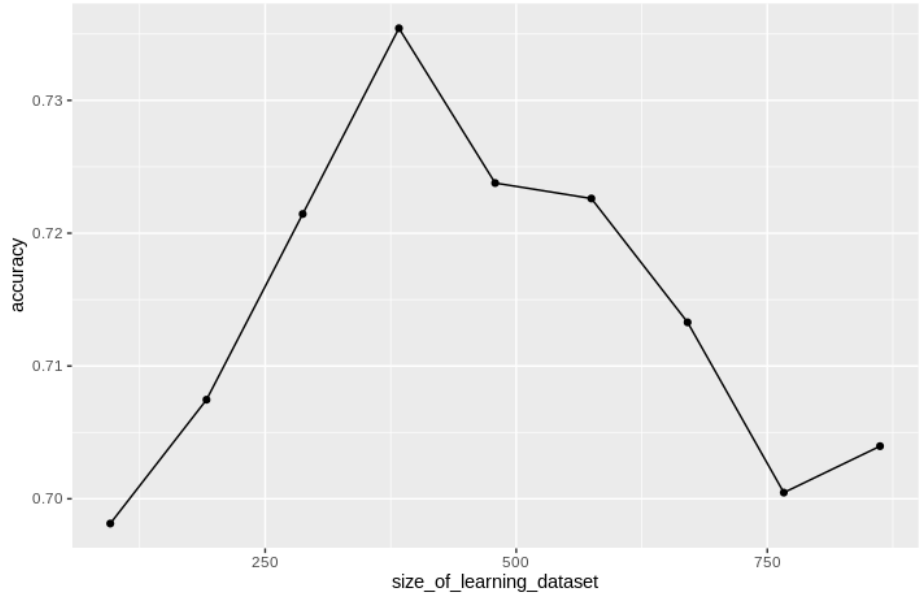

Рис. 1: Зависимость точности классификации от размера обучающей выборки для данных крестики-нолики

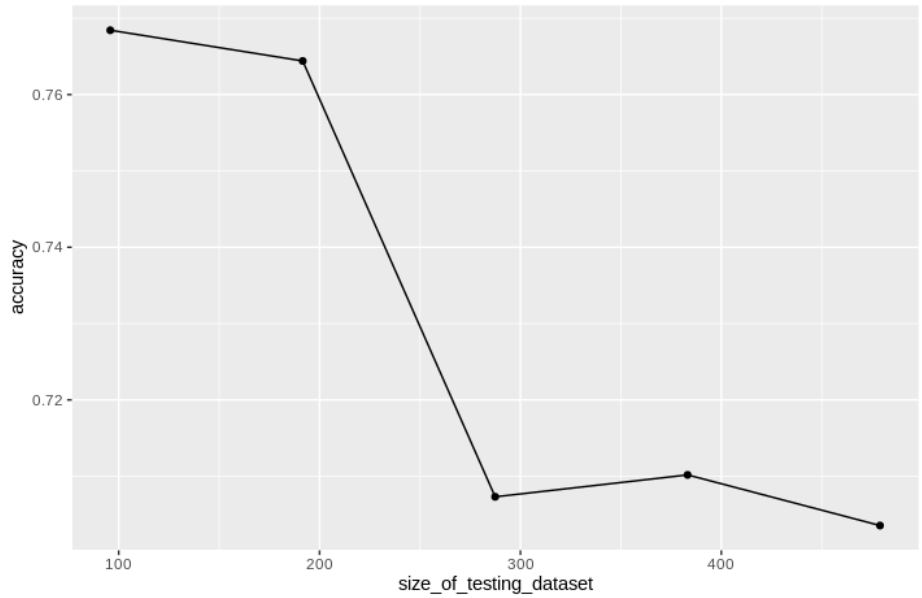

Рис. 2: Зависимость точности классификации от размера тестовой выборки для данных спам

На рис. 3 и 4 представлены графики зависимости точности классификации от размера обучающей и тестовой выборки для данных спам, подтверждающие выводы, сделанные выше.

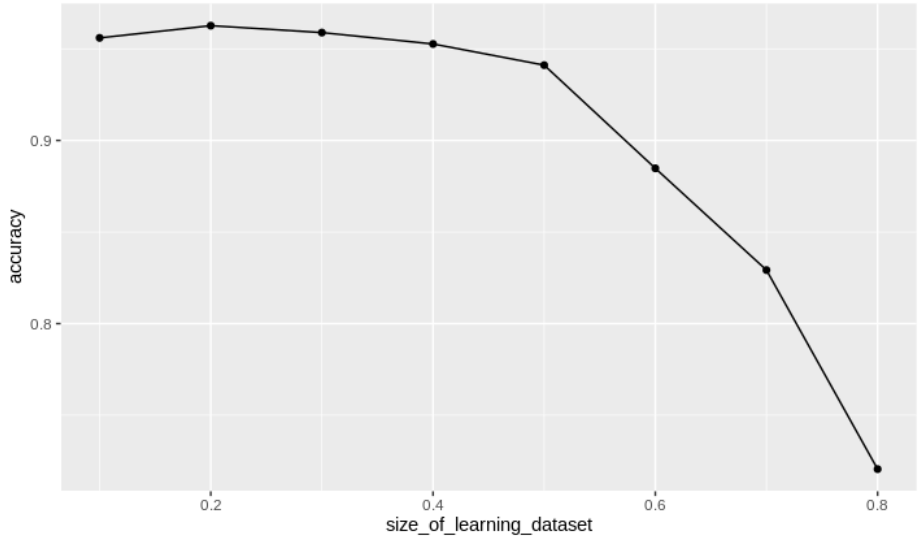

Рис. 3: Зависимость точности классификации от размера обучающей выборки для данных спам

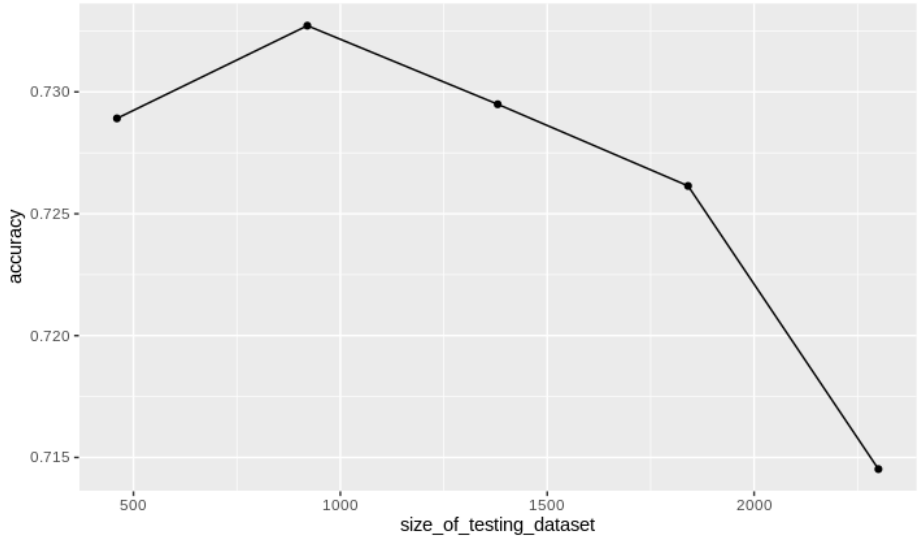

Рис. 4: Зависимость точности классификации от размера тестовой выборки для данных крестики-нолики

2 Задание 2

2.1 Постановка задачи

Сгенерируйте 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.

2.2 Реализация

Диаграмма рассеяния точек представлена на рис. 5. Рис. 6 и 7 показывают гистограммы распределения. Построенный байесовский классификатор предсказывает классы точек с точностью 95%.

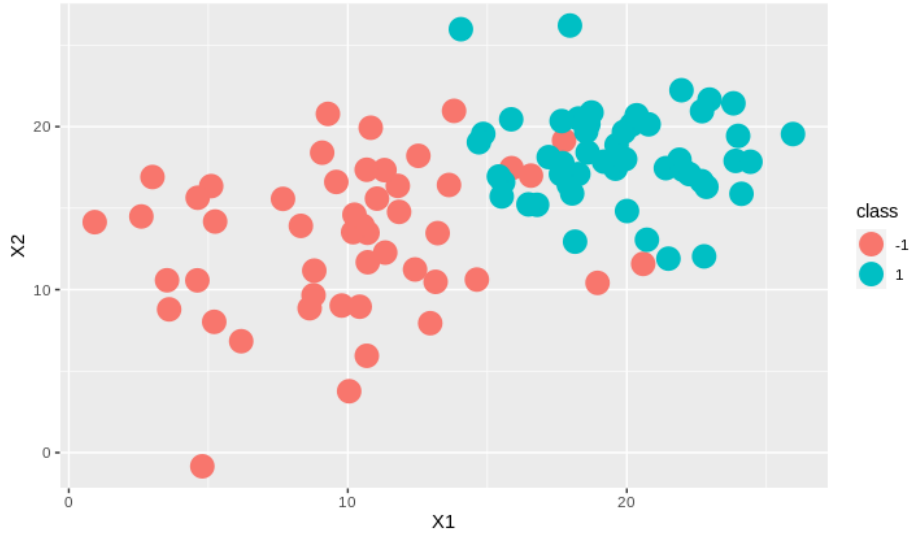

Рис. 5: Диаграмма рассеяния сгенерированных данных

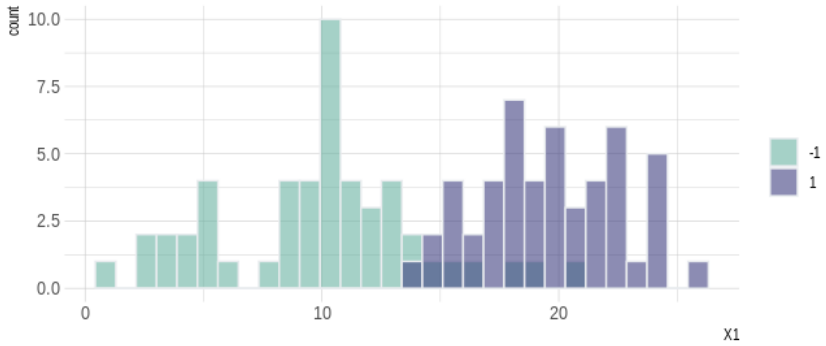

Рис. 6: Гистограмма распределения X1

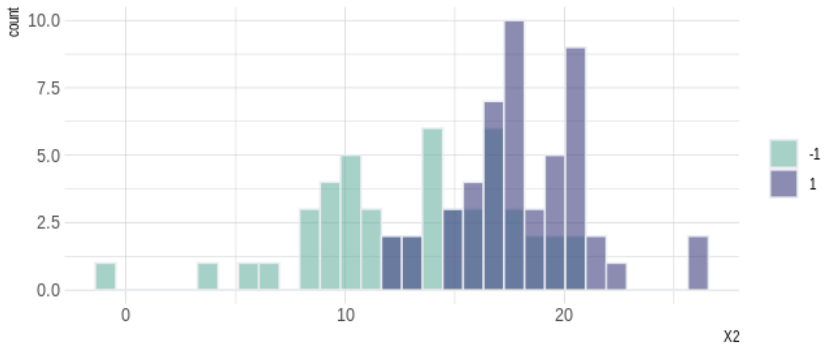

Рис. 7: Гистограмма распределения X2

3 Задание 3

3.1 Постановка задачи

Разработать байесовский классификатор для данных Титаник (Titanic dataset) - https://www.kaggle.com/c/titanic Исходные обучающие данные для классификации – в файле Titanic_train.csv Данные для тестирования – в файле Titanic_test.csv

3.2 Реализация

Были удалены, как не влияющие на выживаемость пассажира признаки столбцы PassengerId, Name, Ticket, Cabin и Embarked.
Данные содержат пропущенные значения (NA). В обучающих данных их всего 117, все они содержатся в поле Age. В данных для тестирования такие значения присутствуют в полях Age (86) и Fare (1). Записи с NA были удалены.
В данных для тестирования отсутствовали метки Survived, поэтому была сформирована валидационная выборка для оценки точности предсказаний модели, составляющая 20% от обучающей выборки.

В итоге обучения на тестовых данных модель предсказала 160 погибших и 172 выживших пассажира. Точность на валидационной выборке составила 72%.

Приложение

Задание 1

```r
library(e1071)
library(ggplot2)

setwd("/home/olga/MyProjects/Polikek/ML/Bayes/datasets")
A_raw <- read.table("Tic_tac_toe.txt", sep = ",", stringsAsFactors = TRUE)
m <- dim(A_raw)[1]
size <- seq(0.1, 0.5, by=0.1)
set.seed(12345)
mean_accuracy <- c()
for (j in 1:length(size)){
  A_rand <- A_raw[ order(runif(m)), ]
  nt <- as.integer(m*size[j])
  accuracy <- c()
  for (k in 1:10){
    A_test <- A_rand[1:nt, ]
    #A_test <- A_rand[(nt+1):n, ]
    A_train <- A_rand[m-400:m, ]
    A_classifier <- naiveBayes(V10 ~ ., data = A_train)
    A_predicted <- predict(A_classifier, A_test)
    accuracy_table <- table(A_predicted, A_test$V10)
    accuracy[k] <- sum(diag(accuracy_table)) / sum(accuracy_table)
  }
  mean_accuracy[j] <- mean(accuracy)
}

new_data <- data.frame(
  size_of_testing_dataset=size * m,
  accuracy=mean_accuracy)
ggplot(new_data, aes(x=size_of_testing_dataset, y=accuracy)) +
  geom_line() +
  geom_point()


library(e1071)
library(ggplot2)
library(kernlab)

data(spam)
size <- seq(0.1, 0.5, by=0.1)
n = dim(spam)[1]
mean_accuracy <- c()
for (j in 1:length(size)){
  accuracy <- c()
  s = n * size[j]
  for (k in 1:10){
    idx <- sample(1:dim(spam)[1], s)
    spamtrain <- spam[-idx, ][1:2000,]
    spamtest <- spam[idx, ]
    model <- naiveBayes(type ~ ., data = spamtrain)
    accuracy_table <- table(predict(model, spamtest), spamtest$type)
    #predict(model, spamtest, type = "raw")
    accuracy[k] <- sum(diag(accuracy_table)) / sum(accuracy_table)
  }
  mean_accuracy[j] <- mean(accuracy)
}

new_data <- data.frame(
  size_of_testing_dataset=size * n,
  accuracy=mean_accuracy)
ggplot(new_data, aes(x=size_of_testing_dataset, y=accuracy)) +
  geom_line() +
  geom_point()
```

Задание 2

```r
library(ggplot2)
library(ggExtra)
library(hrbrthemes)
library(e1071)

data = data.frame(X1 = c(rnorm(50, mean=10, 4), rnorm(50, mean=20, 3)),
                  X2 = c(rnorm(50, mean=14, 4), rnorm(50, mean=18, 3)),
                  class = c(rep("-1",50), rep("1",50)))

ggplot(data, aes(x=X2, fill=class)) +
  geom_histogram( color="#e9ecef", alpha=0.6, position = 'identity') +
  scale_fill_manual(values=c("#69b3a2", "#404080")) +
  theme_ipsum() +
  labs(fill="")

ggplot(data, aes(x=X1, y=X2, color=class)) +
  geom_point(size=6)

rows = sample(nrow(data))
data = data[rows, ]
train = data[1:60,]
test = data[61:100,]
bayes = naiveBayes(class~., train)
prediction = predict(bayes, test)
ac_tab = table(prediction, test$class)
correct = sum(diag(ac_tab))/sum(ac_tab)
```

Задание 3

```r
library(dplyr)
library(ggplot2)
library(rpivotTable)
library(dummies)
setwd("/home/olga/MyProjects/Polikek/ML/Bayes/datasets")

data.train = read.csv("Titanic_train2.csv", sep=",", quote ="\"", stringsAsFactors = TRUE)
data.train = select(data.train, Survived, Pclass, Age, Sex, SibSp, Parch)
data.train = na.omit(data.train)
data.train$Survived = factor(data.train$Survived)
data.train$Pclass = factor(data.train$Pclass, ordered=TRUE, levels = c(3, 2, 1))

data.test = read.csv("Titanic_test2.csv", sep=",", quote ="\"", stringsAsFactors = TRUE)
data.test = select(data.test, Pclass, Age, Sex, SibSp, Parch)
data.test$Pclass = factor(data.test$Pclass, ordered=TRUE, levels = c(3, 2, 1))
data.test = na.omit(data.test)

bayes = naiveBayes(Survived ~ ., data = data.train)
prediction = predict(bayes, data.test)

s = round(seq(dim(data.train)[1]*0.2))
data.valid = data.train[s,]
data.train = data.train[-s,]
bayes = naiveBayes(Survived ~ ., data = data.train)
prediction = predict(bayes, data.valid)
t = table(prediction, data.valid$Survived)
accuracy = sum(diag(t)) / sum(t)
```