Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчёт лабораторная №5.2**

**Дисциплина: Обработка больших данных**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гаранина Л.В.

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Яхонтов А.А.

Краснодар

2025

**Тема:** Байесовская классификация и деревья принятия решений на R.

**Цель**: научиться выполнять классификацию на основе формулы Байеса и деревьев решений.

**Задание:**

1. Добавить найденные в результате кластерного анализа классы, как вектор-столбец в предыдущий DATASET, разделить его на две части (обучающую и тестовую), обучить на обучающей выборке классификатор, а затем применить классификатор к тестовым данным.
2. Для наивного Байесовского классификатора: решить задачу с помощью наивного Байесовского классификатора; проанализировать точность полученных решений для тестовых данных.
3. Для деревьев решений: применить метод деревьев решений для задачи классификации; исследовать дерево решений; если позволяет размерность, построить график; проанализировать точность полученных решений для тестовых данных, сравнить результаты с ранее полученными.
4. Выполнить классификацию с помощью случайного леса, сопоставить результат с результатом дерева решения, прокомментировать результат классификатора; сопоставить результаты с результатами Байесовского классификатора.

Вариант 10: Covid\_Russia.

Фрагмент датасета показан на рисунке 1.



Рисунок 1 - Датасет

**Ход работы**:

Для начала добавим найденные в результате кластерного анализа классы как вектор-столбец в предыдущий датасет и разделим его на 2 части – обучающую (70%) и тестовую (30%).

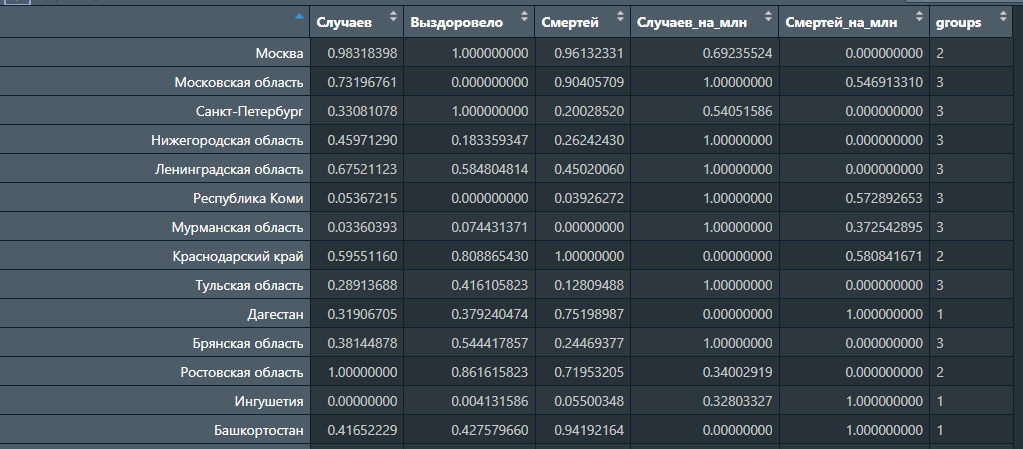


Рисунок 2 – Датасет с классами

Далее решим задачу с помощью наивного Байесовского классификатора. Выведем таблицы вероятностей для каждого критерия. Эти таблицы позволяют оценить, насколько вероятно принадлежность нового объекта к каждому классу и выбирать наиболее вероятный.

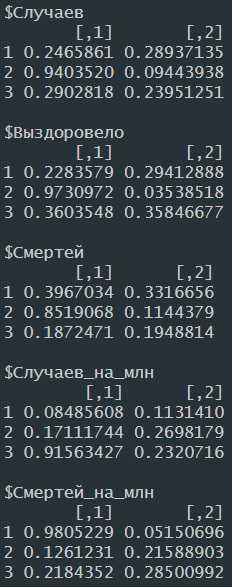


Рисунок 3 – Таблицы вероятностей

Теперь выведем графики по каждому критерию с классами (рисунок 4) и матрицу ошибок (рисунок 5).

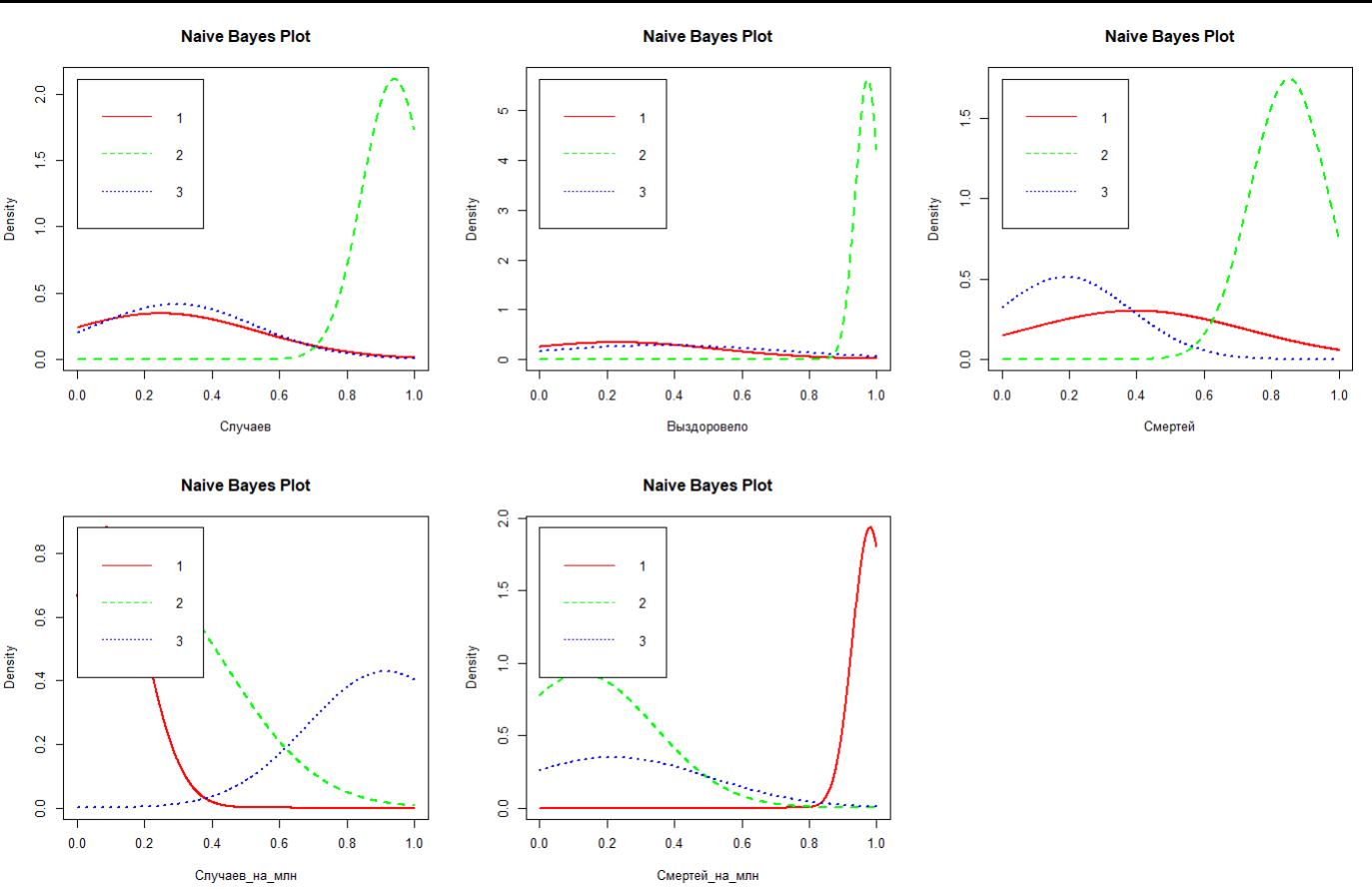


Рисунок 4 – Графики байесовской классификации

Каждая кривая – это распределение значений признака внутри одного кластера. Чем выше кривая в какой-то области, тем чаще там встречаются объекты из соответствующего кластера.

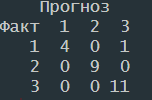


Рисунок 5 – Матрица ошибок

Матрица ошибок сравнивает фактические значения классов, то есть, к какому кластеру действительно принадлежит объект, и предсказанные моделью значения – те, к какому кластеру модель отнесла объект.

По матрице видно, что модель правильно отнесла 4 объекта к кластеру 1, 9 – к 2, 11 – к 3 и 1 объект из кластера 1 был ошибочно отнесен к кластеру 3.

Применим метод деревьев решений для задачи классификации. Будем использовать библиотеку party и выведем само дерево для кластеров.

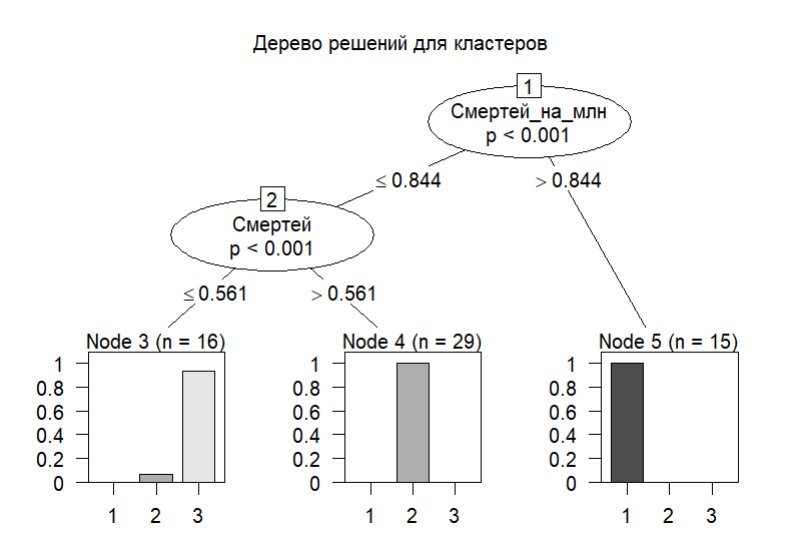


Рисунок 6 – Дерево решений

Дерево решений – модель, которая делит пространство признаков на несколько областей, основываясь на условиях для каждого признака.

Корень дерева – первое условие. В данном случае это смерти на миллион. В зависимости от результата мы переходим в левое или правое поддерево. На листья показана, к какому кластеру относится объект, если его характеристики удовлетворяют всем условиям на пути от корня дерева до этого.

Выведем матрицу ошибок для этого метода.

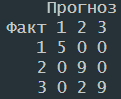
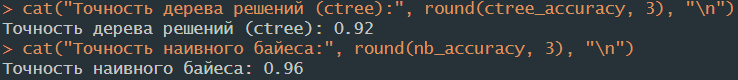


Рисунок 7 –Матрица ошибок для метода деревьев

Видно, что всего 2 объекта из кластера 3 были ошибочно отнесены к кластеру 2.

Сравним также этот метод и Байесовский.

Рисунок 8 – Точности методов

Увидим, что разница между методами совсем небольшая – 0,04.

Выполним классификацию с помощью случайного леса. Также выведем обучение этой модели.

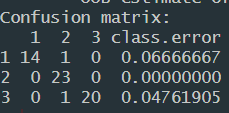


Рисунок 9 – Вывод основных данных модели

Здесь столбцы – фактические классы, а столбцы – предсказанные классы. Последний столбец показывает долю ошибок для каждого класса.

Также выведем матрицу ошибок и точность метода.

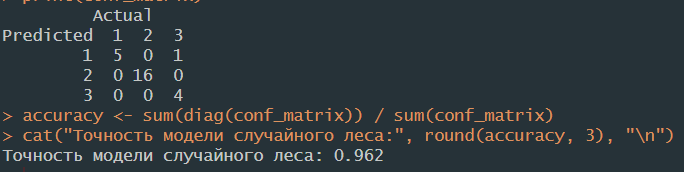


Рисунок 10

Видно, что эта модель самая точная среди всех предыдущих, но не намного.

Посмотрим и на важность признаков.

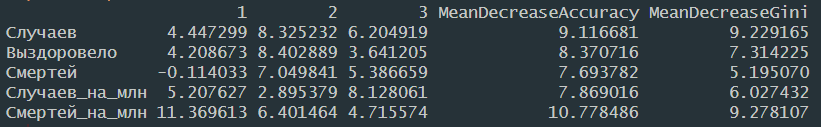


Рисунок 11 – Важности признаков

Здесь выводится важность каждого признака для классификации в модели случайного леса. Есть 2 основные метрики MeanDecreaseAccuracy (показывает, насколько бы уменьшилась точность модели, если бы этот признак был случайным образом перемешан) и MeanDecreaseGini (как часто признак использовался для разделения данных в деревьях).

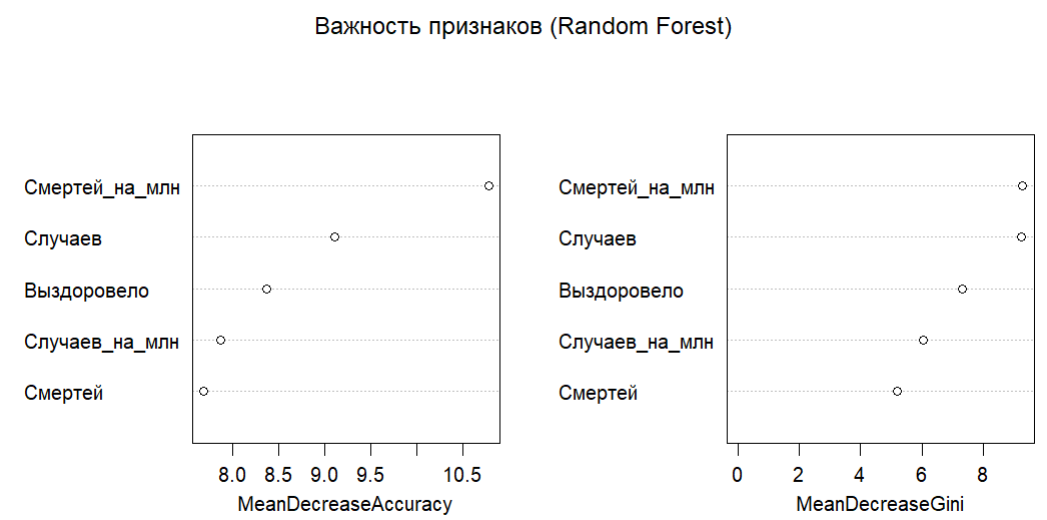


Рисунок 12 – Графики важности признаков

Это графическое представление важности признаков, где визуально можно сравнить, насколько каждый признак важен для модели. Видно, что самые важные признаки это "Смерти на миллион" и "Случаев".

**Вывод:** проделав лабораторную работу, были изучены Байесовская классификация и деревья решений и их применение на данных.

**Листинг программы:**

library(readxl)

setwd("C:/Users/Людмила/Desktop/big data/rstudioLAB")

df <- read\_excel("Covid\_Russia.xlsx", sheet = 1)

names <- df$Город

df <- df[-1]

row.names(df) <- names

library(BBmisc)

df\_scaled = scale(df) # Стандартизация данных

df\_sc\_norm = normalize(df\_scaled, method = "range", range = c(0, 1))

kmeans\_3 <- kmeans(df\_sc\_norm, 3)

# 1. Добавить классы в датафрейм

data\_2\_new <- data.frame(cbind(df\_sc\_norm, factor(kmeans\_3$cluster)))

colnames(data\_2\_new)[ncol(data\_2\_new)] <- "groups"

data\_2\_new$groups <- factor(data\_2\_new$groups)

# Формируем тренировочные и тестовые данные

set.seed(123)

ind <- sample(2, nrow(data\_2\_new), replace = TRUE, prob = c(0.7, 0.3)) # 70% для тренировки и 30% для теста

trainData <- data\_2\_new[ind == 1, ] # Обучающие данные

testData <- data\_2\_new[ind == 2, ] # Тестовые данные

# 2. Решить задачу с помощью наивного Байесовского классификатора

library(klaR)

naive\_model <- NaiveBayes(groups ~ ., data = trainData)

# Таблицы вероятностей для каждого критерия

naive\_model$tables

summary(trainData)

par(mfrow = c(2, 3))

plot(naive\_model, lwd = 2)

# Прогнозирование на тестовых данных

pred <- predict(naive\_model, testData)$class

# Создаём матрицу ошибок

conf\_matrix <- table(Факт = testData$groups, Прогноз = pred)

print(conf\_matrix)

# 3.Применить метод деревьев решений для задачи классификации

library(party)

tree\_ctree <- ctree(groups ~ ., data = trainData)

plot(tree\_ctree, main = "Дерево решений для кластеров")

# Предсказания на тестовых данных

ctree\_pred <- predict(tree\_ctree, newdata = testData)

# Матрица ошибок

ctree\_conf\_matrix <- table(Факт = testData$groups, Прогноз = ctree\_pred)

print(ctree\_conf\_matrix)

# Точность классификации

ctree\_accuracy <- sum(diag(ctree\_conf\_matrix)) / sum(ctree\_conf\_matrix)

cat("Точность дерева решений (ctree):", round(ctree\_accuracy, 3), "\n")

# Сравнение с наивным байесом

nb\_accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)

cat("Точность наивного байеса:", round(nb\_accuracy, 3), "\n")

# 4. Выполнить классификацию с помощью случайного леса

library(randomForest)

rf\_model <- randomForest(groups ~ ., data = trainData, ntree = 100, importance = TRUE)

print(rf\_model)

# Предсказание

rf\_pred <- predict(rf\_model, testData)

# Матрица ошибок и точность

conf\_matrix <- table(Predicted = rf\_pred, Actual = testData$groups)

print(conf\_matrix)

accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)

cat("Точность модели случайного леса:", round(accuracy, 3), "\n")

# Важность признаков

print(importance(rf\_model))

varImpPlot(rf\_model, main = "Важность признаков (Random Forest)")