МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

*Кафедра* *программирования и информационных технологий*

*Классификация по методу kNN*

Отчет по лабораторной работе № *3*

*09.03.02 Информационные системы и технологии*

*Программная инженерия в информационных системах*

Отчёт составил:

Свиридов Фёдор Юрьевич, группа 5.2, вариант 11

Воронеж2023

**Задание:**

1. Выделить обучающую и тестовую выборки.

2. Определить наилучшее значение k.

3. Оценить качество прогноза на тестовой выборке с помощью таблицы сопряженности.

4. Выдать процент ошибок, допущенных классификатором на тестовой выборке.

Содержимое файла 04cars.dat:

Chevrolet Aveo 4dr ;0;0;0;0;0;0;0;11690;10965;1.6;4;103;28;34;2370;98;167;66

Chevrolet Aveo LS 4dr hatch ;0;0;0;0;0;0;0;12585;11802;1.6;4;103;28;34;2348;98;153;66

Chevrolet Cavalier 2dr ;0;0;0;0;0;0;0;14610;13697;2.2;4;140;26;37;2617;104;183;69

Chevrolet Cavalier 4dr ;0;0;0;0;0;0;0;14810;13884;2.2;4;140;26;37;2676;104;183;68

Chevrolet Cavalier LS 2dr ;0;0;0;0;0;0;0;16385;15357;2.2;4;140;26;37;2617;104;183;69

Dodge Neon SE 4dr ;0;0;0;0;0;0;0;13670;12849;2;4;132;29;36;2581;105;174;67

Dodge Neon SXT 4dr ;0;0;0;0;0;0;0;15040;14086;2;4;132;29;36;2626;105;174;67

Ford Focus ZX3 2dr hatch ;0;0;0;0;0;0;0;13270;12482;2;4;130;26;33;2612;103;168;67

Ford Focus LX 4dr ;0;0;0;0;0;0;0;13730;12906;2;4;110;27;36;2606;103;168;67

Ford Focus SE 4dr ;0;0;0;0;0;0;0;15460;14496;2;4;130;26;33;2606;103;168;67

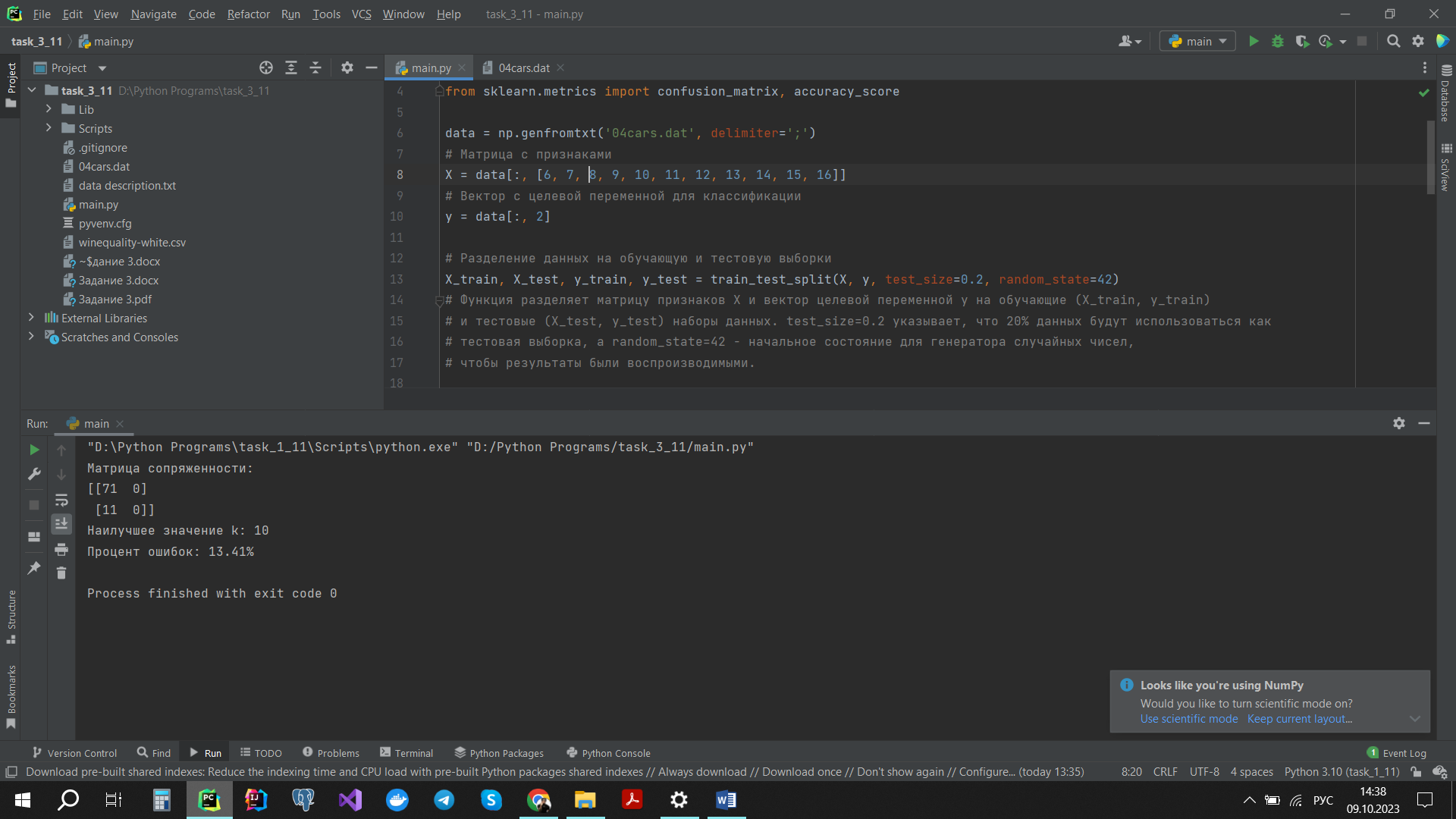
//и так далее

**Код проекта на Python:**

import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.metrics import confusion\_matrix, accuracy\_score  
  
data = np.genfromtxt('04cars.dat', delimiter=';')  
# Матрица с признаками  
X = data[:, [8, 9, 10, 11, 12, 13, 14, 15, 16]]  
# Вектор с целевой переменной для классификации  
y = data[:, 2]  
  
# Разделение данных на обучающую и тестовую выборки  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
# Функция разделяет матрицу признаков X и вектор целевой переменной y на обучающие (X\_train, y\_train)  
# и тестовые (X\_test, y\_test) наборы данных  
  
# Переменные для отслеживания наилучшего значения k и точности  
best\_k = None  
best\_accuracy = 0  
  
# нахождение лучшего значения k  
for k in range(1, 11):  
 knn = KNeighborsClassifier(n\_neighbors=k)  
 # Модель KNN обучается на данных x\_train и y\_train  
 knn.fit(X\_train, y\_train)  
 # Модель используется для предсказания классов на тестовых данных  
 y\_pred = knn.predict(X\_test)  
 # Вычисляется точность модели сравнивая предсказанные значения и истинные  
 accuracy = accuracy\_score(y\_test, y\_pred)  
 if accuracy > best\_accuracy:  
 best\_accuracy = accuracy  
 best\_k = k  
  
# обучение модели с наилучшим значением k  
knn = KNeighborsClassifier(n\_neighbors=best\_k)  
knn.fit(X\_train, y\_train)  
y\_pred = knn.predict(X\_test)  
  
# матрица сопряженности для оценки качества модели  
conf\_matrix = confusion\_matrix(y\_test, y\_pred)  
print("Матрица сопряженности:")  
print(conf\_matrix)  
  
print(f"Наилучшее значение k: {best\_k}")  
# % ошибок = 1 - точность.  
error\_rate = 1 - accuracy\_score(y\_test, y\_pred)  
print(f"Процент ошибок: {error\_rate \* 100:.2f}%")

**Результаты выполнения программы:**

По итогу, наилучшим значением k оказалась равной 10.  
Качество прогноза вышло успешным если сверяться с метриками.  
Процент ошибок составил <13,5%. Все данные выведены в консоли



**Использованные функции и библиотеки:**

NumPy - применяется для математических вычислений: начиная с базовых функций, заканчивая линейной алгеброй

Sklearn:

• model\_selection - позволяет разделить данные на обучающую   
и тестовую выборки

• MDS-neighbors - представляет собой реализацию алгоритма   
K-Nearest Neighbors для классификации

• StandardScaler – используется для оценки качества модели