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

Воронеж 2023

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
```

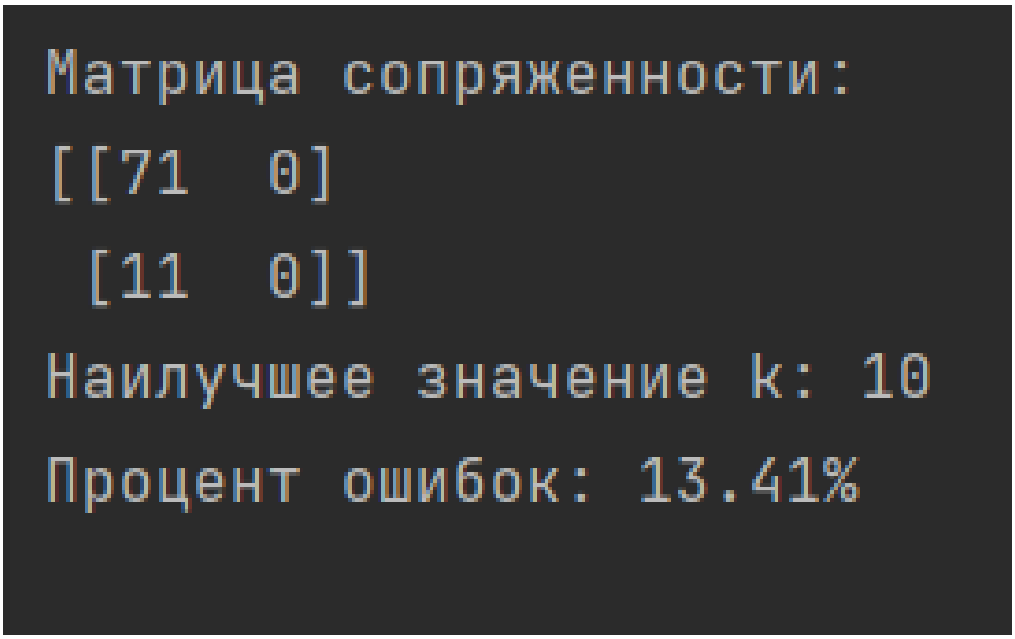
Код проекта на Python:

```
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]]
```

```python
# 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