Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп’ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Звіт

Про виконання лабораторної роботи №3
Дослідницький аналіз даних у Python.
з дисципліни « Методи та системи штучного інтелекту»

Виконав:
Студент групи КН-33
Цьома І.С.

Тернопіль 2024р.

ЗВІТ ЛАБОРАТОРНОЇ РОБОТИ №3

Мета роботи: Набути навики щодо можливостей аналізу даних з використанням класифкатора методом k-найближчих сусідів (k-nn).

Завдання:

Розробити програмну реалізацію Matlab, яка забезпечує виконання наступних кроків для створення KNN класифікатора у Matlab/Python:

- Завантажити базу параметрів квітів iris dataset
- Перемішати записи у завантаженій базі
- Нормалізувати параметри квітів ірису
- Розділити існуючі записи на навчальну і тестові вибірки
- Навчити KNN-класифікатор з різними значеннями К
- Вибрати величину К для найкращих показників якості класифікацій у тестовій вибірці

Хід роботи

1. Завантаження даних

Дані було завантажено з файлу "IrisData_full.csv", який містить 150 записів для трьох видів ірису.

```
data = pd.read_csv("IrisData_full.csv", header=None, names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])
```

2. Перемішування записів у базі

Для усунення впливу порядку записів у базі, було виконано перемішування даних.

```
data = data.sample(frac=1, random_state=42).reset_index(drop=True)
```

3. Нормалізація параметрів квітів

Параметри квітів були нормалізовані, щоб уникнути домінування ознак з великими значеннями над іншими. Для цього було використано *StandardScaler*.

```
scaler = StandardScaler()

X = scaler.fit_transform(data.drop(columns=['species']))

y = data['species']
```

4. Розділення на навчальну і тестову вибірки

Дані були розділені на навчальну та тестову вибірки у співвідношенні 70% на 30%.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

5. Навчання KNN-класифікатора з різними значеннями KKK

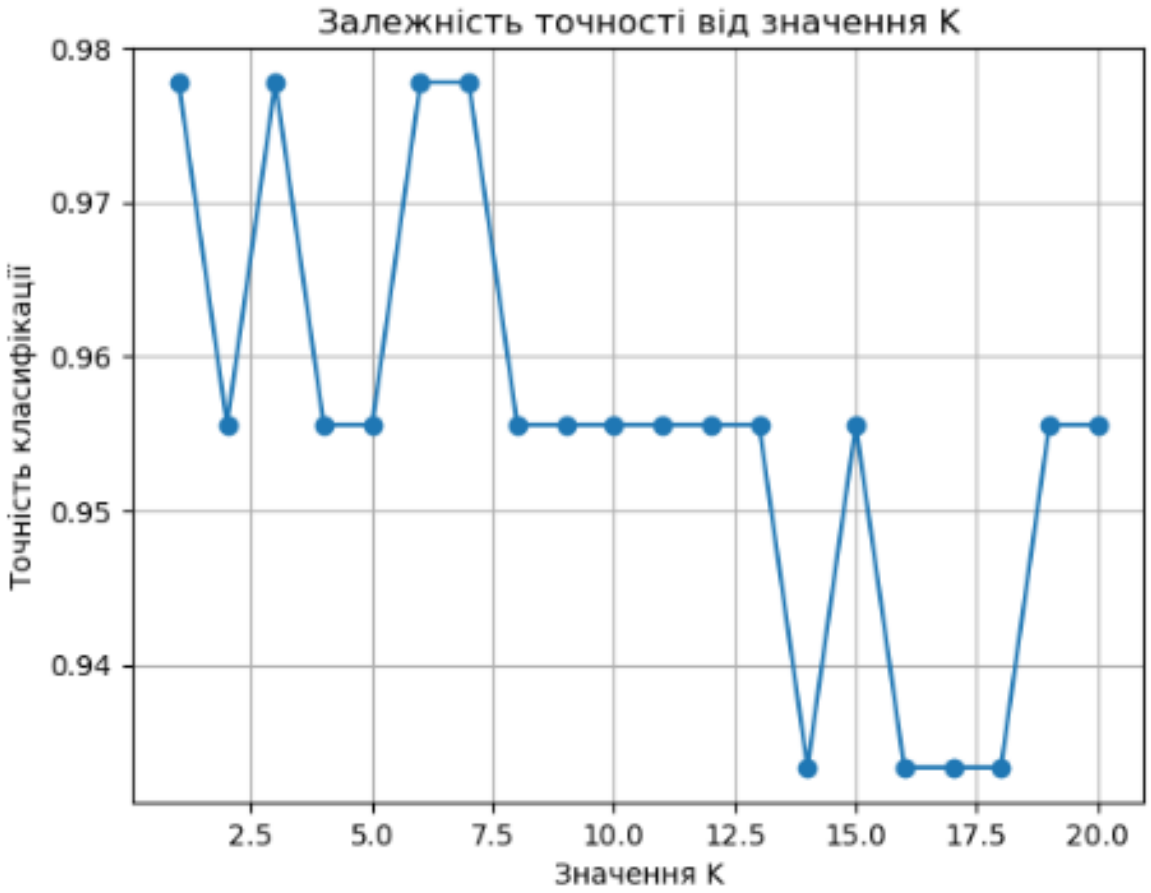
Після налаштування навчальної та тестової вибірок було проведено навчання KNN-класифікатора з різними значеннями KKK від 1 до 20, з метою визначити точність моделі для кожного значення.

```
k_values = range(1, 21)

accuracies = []


for k in k_values:

    knn = KNeighborsClassifier(n_neighbors=k)

    knn.fit(X_train, y_train)

    y_pred = knn.predict(X_test)

    accuracies.append(accuracy_score(y_test, y_pred))
```

6. Вибір оптимального значення KKK

З графіка нижче видно залежність точності класифікації від значення K. Найкраще значення K – це те, яке дає максимальну точність на тестовій вибірці.

Для нашої моделі, оптимальним виявилося значення K = 1, з точністю класифікації ≈ 0.98



Код:

```python
# Імпорт бібліотек
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# 1. Завантаження даних
data = pd.read_csv("IrisData_full.csv", header=None, names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

# 2. Перемішування записів у базі
data = data.sample(frac=1, random_state=42).reset_index(drop=True)

# 3. Нормалізація параметрів квітів
scaler = StandardScaler()
X = scaler.fit_transform(data.drop(columns=['species']))
y = data['species']

# 4. Розділення на навчальну і тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 5. Навчання KNN-класифікатора з різними значеннями K
k_values = range(1, 21)
accuracies = []

for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    accuracies.append(accuracy_score(y_test, y_pred))

# 6. Вибір оптимального значення K
best_k = k_values[np.argmax(accuracies)]
best_accuracy = max(accuracies)

print(f"Найкраще значення K: {best_k}")
print(f"Точність класифікації при K={best_k}: {best_accuracy:.2f}")

# Побудова графіку залежності точності від значення K
plt.plot(k_values, accuracies, marker='o')
plt.xlabel('Значення K')
plt.ylabel('Точність класифікації')
plt.title('Залежність точності від значення K')
plt.grid()
plt.show()
```

Висновки:

1. **Метод нормалізації** параметрів дозволив покращити роботу моделі за рахунок приведення різних ознак до одного масштабу.
2. **Оптимальне значення KKK** для алгоритму KNN було обрано на основі максимального показника точності на тестовій вибірці.
3. **Алгоритм KNN** показав високу точність класифікації квітів ірису з вибраним параметром KKK.

Отримані результати підтверджують, що алгоритм kkk-ближчих сусідів є ефективним методом для класифікаційних задач при правильному виборі гіперпараметрів.