|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА (ИУ12)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**Отчет**

**Название**: Работа с метрическими методами на примере к-ближайших   
соседей (knn)

**Дисциплина:** Введение в искусственный интеллект

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ12-11М |  |  | Д.В. Кузнецов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д.Ю. Евсюков |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цели практики:**

• освоить работу с методом К - ближайших соседей

• освоить умение выбора в методе ближайших соседей параметра К

• овладеть подходами подготовки данных для применения данного метода

**Решение задач**

Импорт необходимых библиотек и подгрузка csv файла

import pandas as pd

from sklearn.model\_selection import KFold, cross\_val\_score

from sklearn.preprocessing import scale

from sklearn.neighbors import KNeighborsClassifier

data = pd.read\_csv('wine.data', header=None)

1. Извлечение из данных признаков и классов

**Решение:**

X = data.iloc[:, 1:] # признаки

y = data.iloc[:, 0] # классы

1. Проведение оценки качества методом кросс-валидации по 5 блокам

**Решение:**

kf = KFold(n\_splits=5, shuffle=True, random\_state=42)

1. Нахождение точности классификации на кросс-валидации для метода k-ближайших соседей при k от 1 до 50.

**Решение:**

results = []

for k in range(1, 51):

    model = KNeighborsClassifier(n\_neighbors=k)

    scores = cross\_val\_score(model, X, y, cv=kf, scoring='accuracy')

    results.append(scores.mean())

optimal\_k = results.index(max(results)) + 1

accuracy\_at\_optimal\_k = max(results)

print(f"Оптимальное значение k: {optimal\_k}")

print(f"Точность на кросс-валидации: {accuracy\_at\_optimal\_k}")

**Ответ**

Оптимальное значение k: 1

Точность на кросс-валидации: 0.7304761904761905

1. Масштабирование признаков.

**Решение:**

X\_scaled = scale(X)

1. Нахождение оптимального параметра и точности после приведения признаков к одному масштабу.

**Решение:**

results\_scaled = []

for k in range(1, 51):

    model = KNeighborsClassifier(n\_neighbors=k)

    scores = cross\_val\_score(model, X\_scaled, y, cv=kf, scoring='accuracy')

    results\_scaled.append(scores.mean())

optimal\_k\_scaled = results\_scaled.index(max(results\_scaled)) + 1

accuracy\_at\_optimal\_k\_scaled = max(results\_scaled)

print(f"Оптимальное значение k (после масштабирования): {optimal\_k\_scaled}")

print(f"Точность на кросс-валидации (после масштабирования): {accuracy\_at\_optimal\_k\_scaled}")

**Ответ:**

Оптимальное значение k (после масштабирования): 29

Точность на кросс-валидации (после масштабирования): 0.9776190476190475

**Вывод**

В результате проведенного анализа вин с использованием метода k ближайших соседей (kNN), были получены следующие выводы:

Без масштабирования признаков:

Оптимальное значение k: 1

Точность на кросс-валидации: 73.05%

При использовании метода kNN без масштабирования признаков удалось достичь точности около 73%.

После масштабирования признаков:

Оптимальное значение k: 29

Точность на кросс-валидации: 97.76%

Масштабирование признаков привело к значительному улучшению модели, с оптимальным значением k равным 29 и точностью на кросс-валидации близкой к 98%.

Масштабирование признаков оказало существенное влияние на качество модели kNN для данного датасета, позволив достичь высокой точности предсказаний. Учет рекомендаций по выбору оптимального значения k, а также стандартизации переменных, является важным этапом при построении и настройке моделей машинного обучения.