|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Калужский филиал федерального государственного автономного образовательного учреждения высшего образования**  **«Московский государственный технический университет  имени Н.Э. Баумана  (национальный исследовательский университет)»**  **(КФ МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК4 Программное обеспечение ЭВМ, информационные технологии

**ПРАКТИЧЕСКАЯ РАБОТА**

**«Линейные классификаторы и их свойства»**

**по дисциплине: «Методы машинного обучения»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил: студент группы ИУК4-72Б | |  |  | Губин Е.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил: | |  |  | Семененко М.Г. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | | | |
|  | - Балльная оценка:  - Оценка: | | | |

Калуга, 2025

1. **Линейная и квадратичная регрессия**

* Построить линейную и квадратичную регрессию по данным.
* Для каждой модели рассчитать предсказанные значения (Линейная Y, Квадратичная Y) для каждого профиля.
* Вычислить среднюю квадратичную ошибку (MSE) и сумму квадратов ошибок (SSE) для каждой модели.
* Построить точечную диаграмму исходных данных с наложением линий регрессии.

1. **Аппроксимация функции Рунге**

* Аппроксимировать функцию полиномами 5-й и 6-й степени.
* Рассчитать предсказанные значения функции для этих полиномов.
* Вычислить среднюю ошибку (MSE) и сумму квадратов ошибок (SSE) для каждой аппроксимации.
* Построить графики функции Рунге и аппроксимационных полиномов с точками.
* Составить таблицу с исходными значениями функции и предсказаниями полиномов.

**Листинг программы:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

profiles = ["A", "B", "C", "D", "E", "F", "G", "H", "P", "J", "K"]

x = np.array([4.3, 3.9, 5.0, 4.3, 3.7, 4.4, 3.8, 3.1, 4.7, 4.2, 3.4])

y = np.array([90, 84, 67, 83, 89, 89, 89, 83, 73, 85, 84])

coeffs\_lin = np.polyfit(x, y, 1)

lin\_eq = np.poly1d(coeffs\_lin)

coeffs\_quad = np.polyfit(x, y, 2)

quad\_eq = np.poly1d(coeffs\_quad)

y\_pred\_lin = lin\_eq(x)

y\_pred\_quad = quad\_eq(x)

mse\_lin = np.mean((y - y\_pred\_lin) \*\* 2)

mse\_quad = np.mean((y - y\_pred\_quad) \*\* 2)

sse\_lin = np.sum((y - y\_pred\_lin) \*\* 2)

sse\_quad = np.sum((y - y\_pred\_quad) \*\* 2)

results\_reg = pd.DataFrame({

    "Profile": profiles,

    "X (Преподаватели)": x,

    "Y (Экзамен)": y,

    "Линейная Y": y\_pred\_lin.round(2),

    "Квадратичная Y": y\_pred\_quad.round(2)

})

print("=== Результаты регрессии ===")

print(results\_reg)

print("\nОшибки:")

print(*f*"Линейная: MSE = {mse\_lin*:.3f*}, SSE = {sse\_lin*:.3f*}")

print(*f*"Квадратичная: MSE = {mse\_quad*:.3f*}, SSE = {sse\_quad*:.3f*}")

plt.scatter(x, y, *label*="Данные", *color*="blue")

x\_line = np.linspace(min(x), max(x), 200)

plt.plot(x\_line, lin\_eq(x\_line), "r-", *label*=*f*"Линейная: {coeffs\_lin[0]*:.2f*}x+{coeffs\_lin[1]*:.2f*}")

plt.plot(x\_line, quad\_eq(x\_line), "g--", *label*=*f*"Квадратичная: {coeffs\_quad[0]*:.2f*}x²+{coeffs\_quad[1]*:.2f*}x+{coeffs\_quad[2]*:.2f*}")

plt.xlabel("Средняя оценка преподавателям")

plt.ylabel("Средняя экзаменационная оценка")

plt.legend()

plt.title("Линейная и квадратичная регрессия")

plt.grid()

plt.show()

*def* runge(*x*):

    return 1 / (1 + 25 \* *x*\*\*2)

X\_runge = np.linspace(-2, 2, 15)

Y\_runge = runge(X\_runge)

coeffs\_5 = np.polyfit(X\_runge, Y\_runge, 5)

coeffs\_6 = np.polyfit(X\_runge, Y\_runge, 6)

poly5 = np.poly1d(coeffs\_5)

poly6 = np.poly1d(coeffs\_6)

Y\_pred\_5 = poly5(X\_runge)

Y\_pred\_6 = poly6(X\_runge)

mse\_5 = np.mean((Y\_runge - Y\_pred\_5) \*\* 2)

mse\_6 = np.mean((Y\_runge - Y\_pred\_6) \*\* 2)

sse\_5 = np.sum((Y\_runge - Y\_pred\_5) \*\* 2)

sse\_6 = np.sum((Y\_runge - Y\_pred\_6) \*\* 2)

results\_runge = pd.DataFrame({

    "X": X\_runge.round(2),

    "Runge f(x)": Y\_runge.round(5),

    "Полином 5 ст.": Y\_pred\_5.round(5),

    "Полином 6 ст.": Y\_pred\_6.round(5)

})

print("\n=== Приближение функции Рунге ===")

print(results\_runge)

print("\nОшибки для Рунге:")

print(*f*"Полином 5-й ст.: MSE = {mse\_5*:.6f*}, SSE = {sse\_5*:.6f*}")

print(*f*"Полином 6-й ст.: MSE = {mse\_6*:.6f*}, SSE = {sse\_6*:.6f*}")

x\_plot = np.linspace(-2, 2, 200)

plt.plot(x\_plot, runge(x\_plot), "k-", *label*="Функция Рунге")

plt.plot(x\_plot, poly5(x\_plot), "r--", *label*="Полином 5-й степени")

plt.plot(x\_plot, poly6(x\_plot), "b-.", *label*="Полином 6-й степени")

plt.scatter(X\_runge, Y\_runge, *color*="black", *zorder*=5, *label*="Точки")

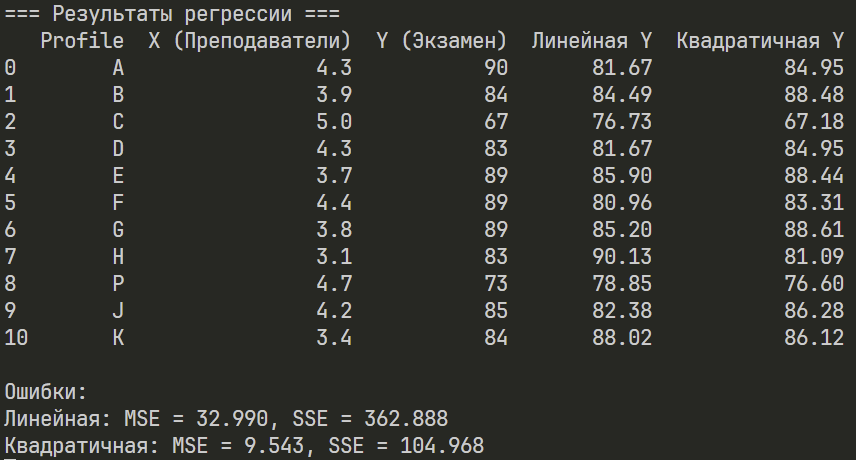
plt.legend()

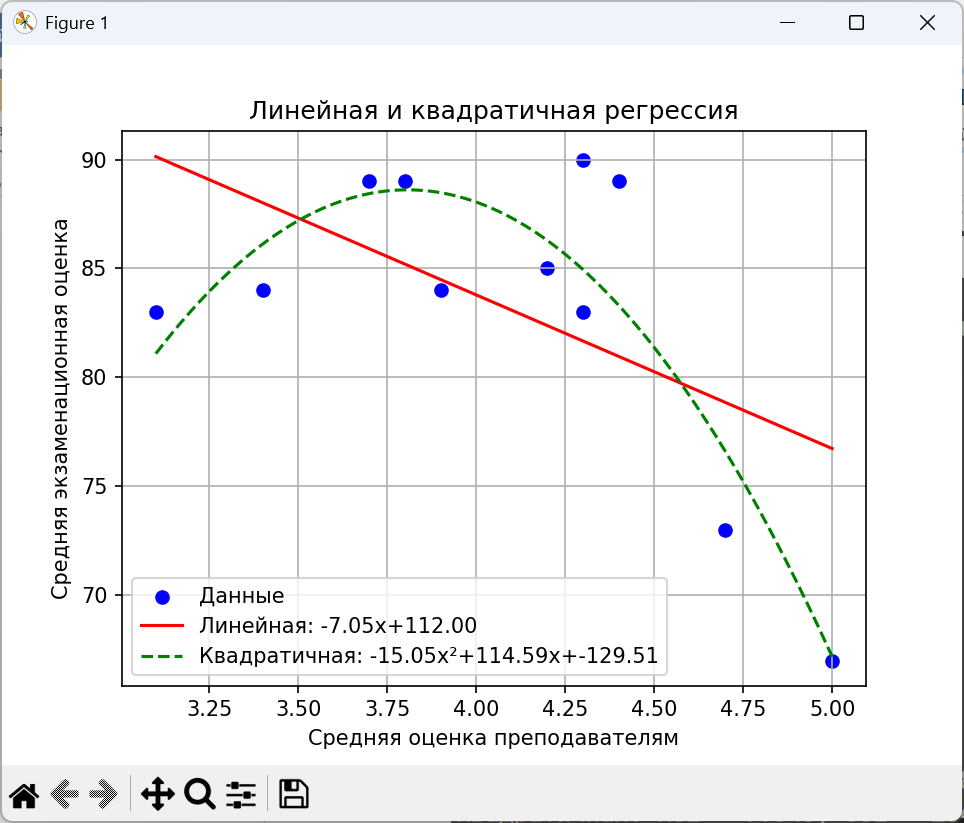
plt.title("Аппроксимация функции Рунге")

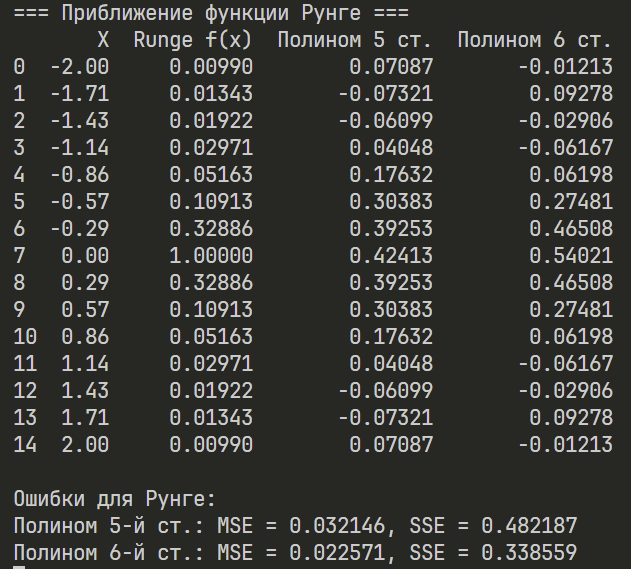
plt.grid()

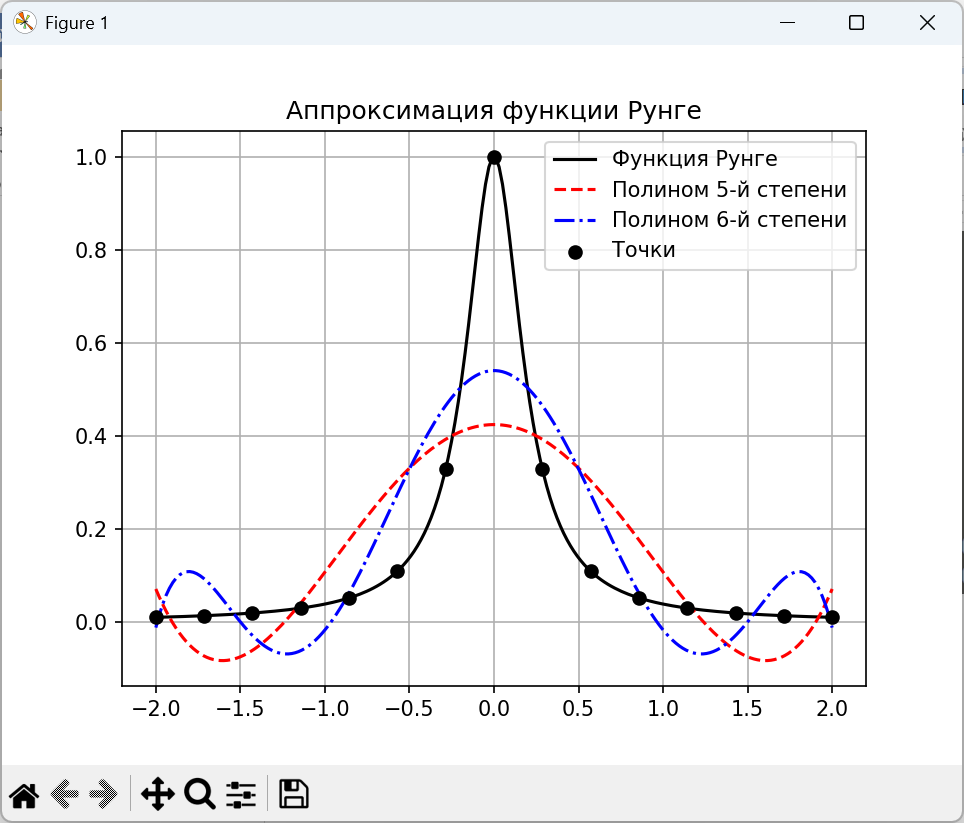
plt.show()

**Результаты выполнения программы:**









**Вывод:** в ходе лабораторной работы получены навыки по созданию модели, предсказывающую экзаменационные оценки и вычислению ошибок этой модели.