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»

по дисциплине: «Методы машинного обучения»

Выполнил: студент группы ИУК4-72Б ____________ Моряков В.Ю.
(Подпись) (И.О. Фамилия)

Проверил: ____________ Семененко М.Г.
(Подпись) (И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2025

Цель: Изучить процессы аппроксимации функций с помощью полиномиальных моделей различной степени и проанализировать явление переобучения. Научиться вычислять градиенты и гессианы функций нескольких переменных, а также находить и визуализировать точки минимума с помощью аналитических и графических методов.

Задачи:

Задача 1

Используя функцию Рунге $y = 1/(1 + 25x^2)$, сформировать обучающую выборку по правилу

$$X^\ell = \left\{x_i = 4\tfrac{i-1}{\ell-1} - 2 \;\middle|\; i = 1, \dots, \ell\right\}.$$

Сформировать контрольную выборку по правилу

$$X^k = \left\{x_i = 4\tfrac{i-0.5}{\ell-1} - 2 \;\middle|\; i = 1, \dots, \ell - 1\right\}.$$

Объем выборки $l = 15$. По обучающей выборке выбрать оптимальную степень полинома

$$a(x, \theta) = \theta_0 + \theta_1 x + \dots + \theta_n x^n$$

соответствующую минимальной квадратичной функции потерь, задавая вручную степень полинома. Для контрольной выборки построить полином выбранной степени и расситать значение квадратичной функции потерь. Построить график.

Задача 2

Допустим, что задана решающая функция линейного классификатора $f(x_1, x_2)$.
Найти координаты и значение функции в точке минимума, используя встроенные функции (пример решения задачи в системе WolframAlpha показан на рисунке). Построить график функции с точкой решения. Вид функции задан в таблице. Номер варианта – номер по списку.

Вариант 1 $f(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 5)^2$

Результаты выполнения программы:

Задание 1

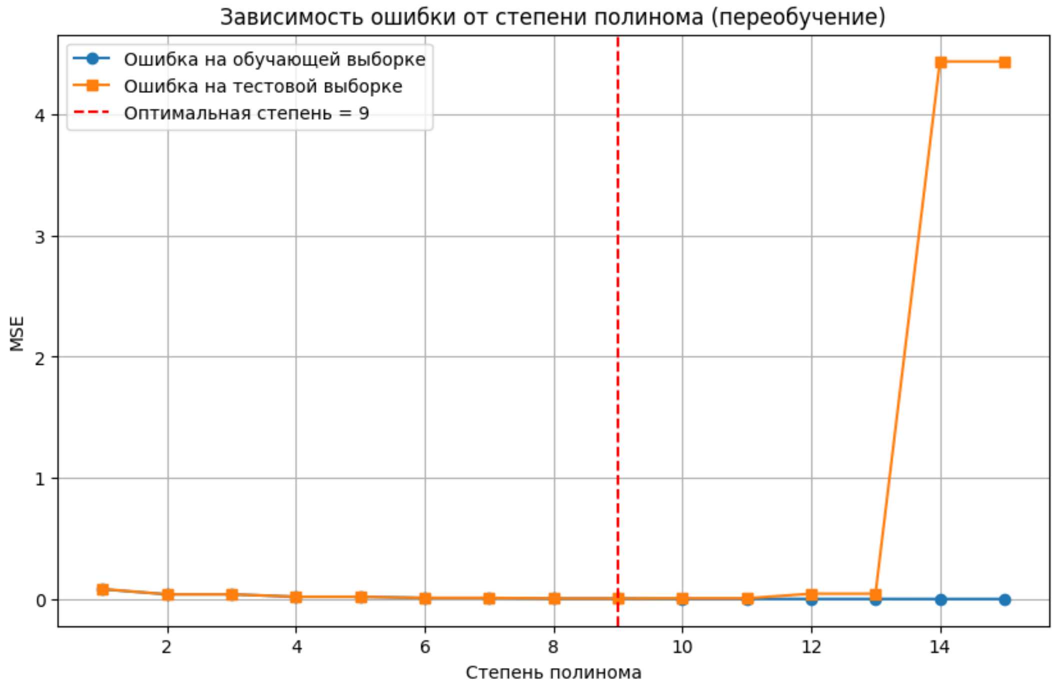

Рисунок 1 — График зависимости ошибки от степени полинома

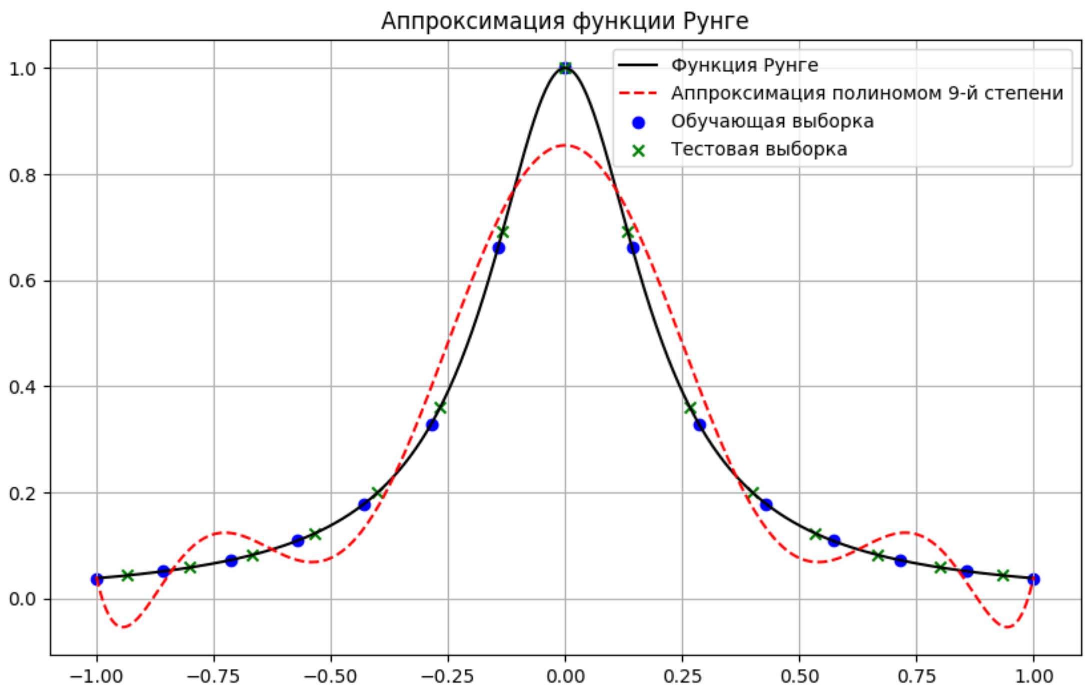

Рисунок 2 — Аппроксимация функции врунге

```
Коэффициенты полинома 1: [8.70824274e-18 2.58903983e-01]
Коэффициенты полинома 2: [-6.02263994e-01 -1.85775845e-16  4.88337886e-01]
Коэффициенты полинома 3: [ 4.27037156e-16 -6.02263994e-01 -3.71551690e-16  4.88337886e-01]
Коэффициенты полинома 4: [ 1.52385748e+00 -5.01077447e-16 -2.07280867e+00  1.02287319e-16
  6.52845948e-01]
Коэффициенты полинома 5: [ 1.88107935e-15  1.52385748e+00 -1.50323234e-15 -2.07280867e+00
  1.85775845e-16  6.52845948e-01]
Коэффициенты полинома 6: [-4.22098005e+00  1.68348063e-15  7.82791860e+00  9.68232832e-16
 -4.35726170e+00  1.91018598e-16  7.70264002e-01]
Коэффициенты полинома 7: [-2.77715531e-14 -4.22098005e+00  4.43316566e-14  7.82791860e+00
 -1.90409430e-14 -4.35726170e+00  2.41508599e-15  7.70264002e-01]
Коэффициенты полинома 8: [ 1.33727155e+01 -6.39503955e-14 -3.04569742e+01  5.16412254e-14
  2.35838705e+01 -1.91807100e-14 -7.31204912e+00  1.94112514e-14
  8.54202123e-01]
Коэффициенты полинома 9: [-2.72210789e-13  1.33727155e+01  5.02129032e-13 -3.04569742e+01
 -3.12005077e-13  2.35838705e+01  7.91702366e-14 -7.31204912e+00
 -5.57327535e-15  8.54202123e-01]
Коэффициенты полинома 10: [-5.21138258e+01  1.50942861e-13  1.35960555e+02 -2.18527759e-13
 -1.29349955e+02  5.38829125e-14  5.54494681e+01  5.86508426e-14
 -1.08225745e+01 -2.23146807e-14  9.14469393e-01]
Коэффициенты полинома 11: [-2.37976432e-12 -5.21138258e+01  6.14483517e-12  1.35960555e+02
 -5.96396652e-12 -1.29349955e+02  2.53195487e-12  5.54494681e+01
 -4.24913675e-13 -1.08225745e+01  1.93206879e-14  9.14469393e-01]
Коэффициенты полинома 12: [ 2.88497306e+02 -3.52937549e-11 -8.15210960e+02  9.06775923e-11
  8.73608257e+02 -8.26071881e-11 -4.49365121e+02  3.13576722e-11
...
Коэффициенты полинома 15: [ 1.77103573e-08 -3.74491696e+03 -5.37339849e-08  1.08495594e+04
  6.39683474e-08 -1.20727039e+04 -3.84607985e-08  6.58241591e+03
  1.24608062e-08 -1.87140780e+03 -2.10005238e-09  2.77585302e+02
  1.55010781e-10 -2.14935253e+01 -4.72615054e-12  1.00000000e+00]
```

Рисунок 3 — Коэффициенты полиномов

Листинг результата:

```
Оптимальная степень полинома по тестовой выборке: 9
MSE на обучающей выборке: 0.003293
MSE на тестовой выборке: 0.004523
```

Задание 2

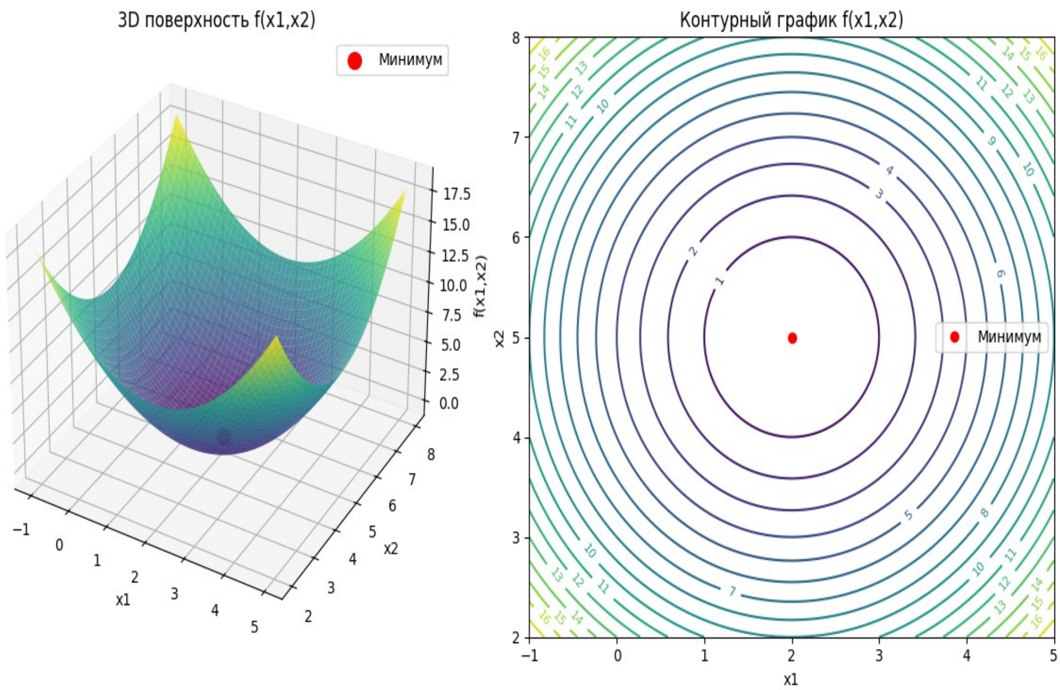

Рисунок 4 — Минимум функции модели на python

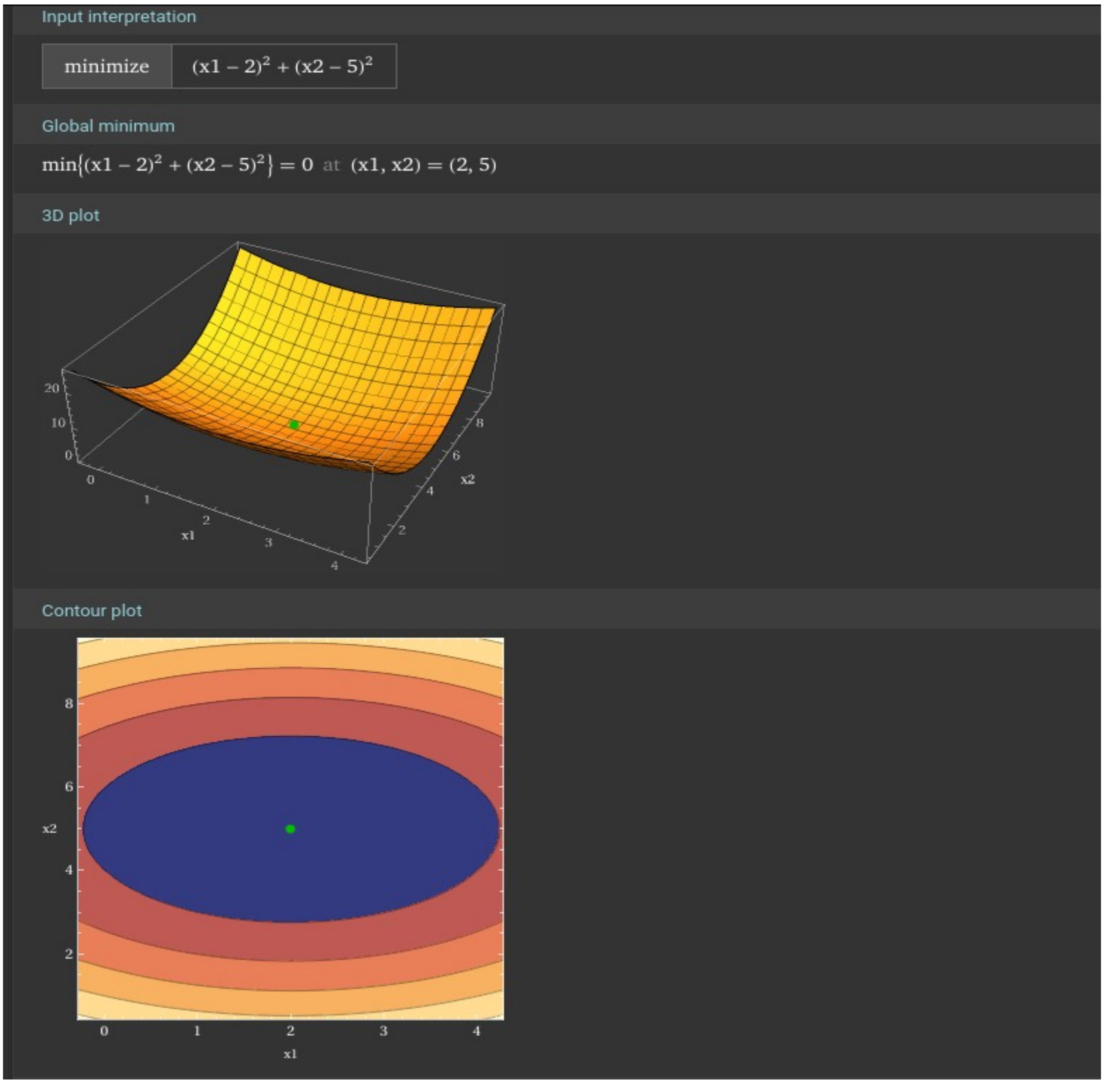

Рисунок 5 — Минимум функции wf-alpha

Листинг результата:

```
Координаты минимума: (2, 5)
Значение функции в минимуме: 0
Гессиан: Matrix([[2, 0], [0, 2]])
```

Листинг программы:

```
# %% [markdown]
# # Библиотеки

# %%
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sympy as sp

# %% [markdown]
# # Задача 1

# %%
def runge_function(x):
    return 1 / (1 + 25 * x**2)

def generate_samples(l=15):
    X_train = np.linspace(-1, 1, l)
```

```
    Y_train = runge_function(X_train)

    step = 2 / l
    X_test = np.linspace(-1 + step/2, 1 - step/2, l)
    Y_test = runge_function(X_test)

    return X_train, Y_train, X_test, Y_test

def fit_polynomials(X_train, Y_train, X_test, Y_test, max_degree=15):
    results = {}
    for deg in range(1, max_degree+1):
        coeffs = np.polyfit(X_train, Y_train, deg)
        print(f"Коэффициенты полнинома {deg}: {coeffs}")
        poly = np.poly1d(coeffs)

        Y_train_pred = poly(X_train)
        Y_test_pred = poly(X_test)

        mse_train = np.mean((Y_train - Y_train_pred) ** 2)
        mse_test = np.mean((Y_test - Y_test_pred) ** 2)

        results[deg] = {
            "poly": poly,
            "mse_train": mse_train,
            "mse_test": mse_test
        }
    return results

def plot_mse(results):
    degrees = list(results.keys())
    mse_train = [results[d]["mse_train"] for d in degrees]
    mse_test = [results[d]["mse_test"] for d in degrees]

    best_degree = min(results, key=lambda d: results[d]["mse_test"])

    plt.figure(figsize=(10, 6))
    plt.plot(degrees, mse_train, "o-", label="Ошибка на обучающей выборке")
    plt.plot(degrees, mse_test, "s-", label="Ошибка на тестовой выборке")
    plt.axvline(best_degree, color="red", linestyle="--", label=f"Оптимальная степень = {best_degree}")
    plt.xlabel("Степень полинома")
    plt.ylabel("MSE")
    plt.title("Зависимость ошибки от степени полинома (переобучение)")
    plt.legend()
    plt.grid()
    plt.show()

    return best_degree

def plot_results(X_train, Y_train, X_test, Y_test, poly):
    x_plot = np.linspace(-1, 1, 500)
    plt.figure(figsize=(10, 6))
    plt.plot(x_plot, runge_function(x_plot), "k-", label="Функция Рунге")
    plt.plot(x_plot, poly(x_plot), "r--", label=f"Аппроксимация полиномом {poly.order}-й степени")
    plt.scatter(X_train, Y_train, color="blue", label="Обучающая выборка")
    plt.scatter(X_test, Y_test, color="green", marker="x", label="Тестовая выборка")
    plt.legend()
    plt.grid()
    plt.title("Аппроксимация функции Рунге")
    plt.show()

X_train, Y_train, X_test, Y_test = generate_samples(l=15)
results = fit_polynomials(X_train, Y_train, X_test, Y_test, max_degree=15)

best_degree = plot_mse(results)

best_poly = results[best_degree]["poly"]
mse_train = results[best_degree]["mse_train"]
mse_test = results[best_degree]["mse_test"]

print(f"Оптимальная степень полинома по тестовой выборке: {best_degree}")
print(f"MSE на обучающей выборке: {mse_train:.6f}")
print(f"MSE на тестовой выборке: {mse_test:.6f}")
```

```
plot_results(X_train, Y_train, X_test, Y_test, best_poly)

x1, x2 = sp.symbols('x1 x2', real=True)
f = (x1-2)**2 + (x2-5)**2
grad = [sp.diff(f, v) for v in (x1, x2)]
crit = sp.solve(grad, (x1, x2))
val = sp.N(f.subs({x1:2, x2:5}))
H = sp.hessian(f, (x1, x2))

print("Координаты минимума: (2, 5)")
print("Значение функции в минимуме:", val)
print("Гессиан:\n", H)


# %% [markdown]
# # Задача 2

# %%
x1, x2 = sp.symbols('x1 x2', real=True)
f = (x1-2)**2 + (x2-5)**2

grad = [sp.diff(f, v) for v in (x1, x2)]
crit = sp.solve(grad, (x1, x2))
val = sp.N(f.subs({x1:2, x2:5}))
H = sp.hessian(f, (x1, x2))

print("Координаты минимума:", crit)
print("Значение функции в минимуме:", val)
print("Гессиан:\n", H)

f_numeric = sp.lambdify((x1, x2), f, "numpy")

x1_vals = np.linspace(-1, 5, 100)
x2_vals = np.linspace(2, 8, 100)
X1, X2 = np.meshgrid(x1_vals, x2_vals)
Z = f_numeric(X1, X2)

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(121, projection='3d')
ax.plot_surface(X1, X2, Z, cmap='viridis', alpha=0.8)
ax.scatter(2, 5, 0, color='red', s=100, label='Минимум')
ax.set_xlabel('x1')
ax.set_ylabel('x2')
ax.set_zlabel('f(x1,x2)')
ax.set_title('3D поверхность f(x1,x2)')
ax.legend()

ax2 = fig.add_subplot(122)
contours = ax2.contour(X1, X2, Z, levels=20, cmap='viridis')
ax2.clabel(contours, inline=True, fontsize=8)
ax2.plot(2, 5, 'ro', label='Минимум')
ax2.set_xlabel('x1')
ax2.set_ylabel('x2')
ax2.set_title('Контурный график f(x1,x2)')
ax2.legend()

plt.tight_layout()
plt.show()
```

Вывод: В ходе работы была исследована аппроксимация функции Рунге полиномами различной степени и проанализировано явление переобучения. Найдена оптимальная степень полинома по минимальной тестовой ошибке. Также для функции двух переменных с помощью градиента и гессиана определена точка минимума и подтвержден её характер графически.