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 ____________ 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

Цель: изучить применение линейных классификаторов для аппроксимации функции Рунге методом полиномиальной регрессии.

Задачи:

1. Сформировать обучающую и контрольную выборки, построить полиномиальные модели разной степени и выбрать оптимальную по функции потерь.
2. Оценить качество выбранной модели на контрольной выборке и визуализировать результат аппроксимации.

Задание №1:

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

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

Сформировать контрольную выборку по правилу:

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

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 рассчитать значение квадратичной функции потерь. Построить график.

Результаты выполнения программы:

```
Степень 1: MSE(train)=0.063188, MSE(test)=0.047771
Степень 2: MSE(train)=0.045230, MSE(test)=0.030814
Степень 3: MSE(train)=0.045230, MSE(test)=0.030814
Степень 4: MSE(train)=0.032146, MSE(test)=0.019696
Степень 5: MSE(train)=0.032146, MSE(test)=0.019696
Степень 6: MSE(train)=0.022571, MSE(test)=0.012575
Степень 7: MSE(train)=0.022571, MSE(test)=0.012575
Степень 8: MSE(train)=0.015447, MSE(test)=0.009020
Степень 9: MSE(train)=0.015447, MSE(test)=0.009020
Степень 10: MSE(train)=0.009979, MSE(test)=0.015833
```

Рисунок 1 Степени полиномов и ошибки по ним

В результате расчётов видно, что минимальную ошибку имеет полином 8-й степени, его и возьмём за самый оптимальный полином. Выполнена аппроксимация функции Рунге полиномом 8-й степени. Результаты аппроксимации отображены на рис. 2.

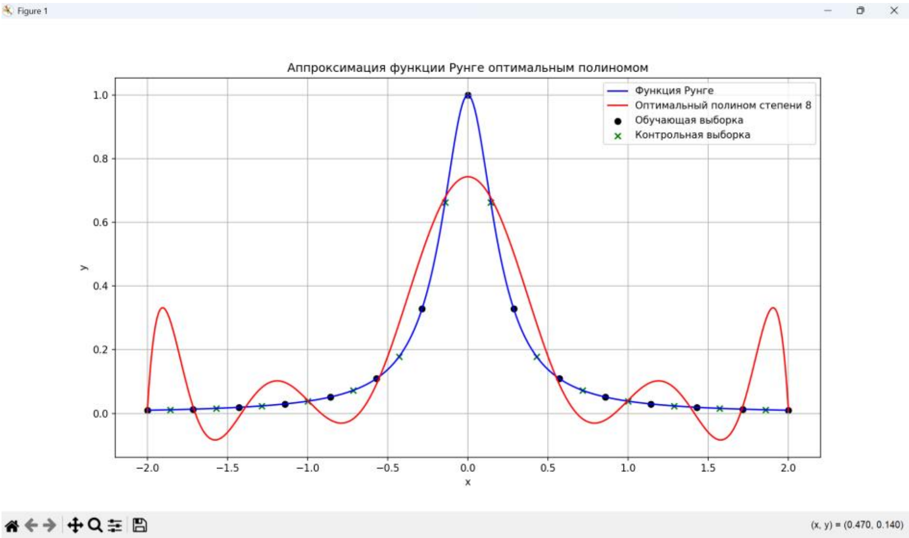
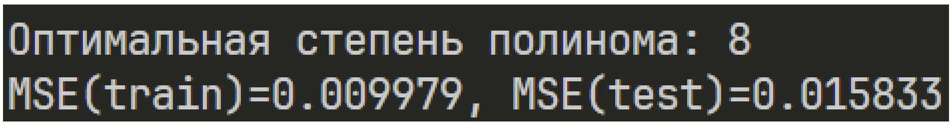


Рисунок 2 Аппроксимация функции Рунге



```
Оптимальная степень полинома: 8
MSE(train)=0.009979, MSE(test)=0.015833
```

Рисунок 3 Оптимальная степень полинома и ошибка

Листинг программы:

```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import PolynomialFeatures

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error


def runge_function(x):

    return 1 / (1 + 25 * x**2)


l = 15

X_train = np.array([4 * ((i - 1) / (l - 1)) - 2 for i in range(1, l + 1)])

y_train = runge_function(X_train)

X_test = np.array([4 * ((i - 0.5) / (l - 1)) - 2 for i in range(1, l)])

y_test = runge_function(X_test)


max_degree = 10

results = []
```

```
for degree in range(1, max_degree + 1):
    poly = PolynomialFeatures(degree=degree)
    X_poly = poly.fit_transform(X_train.reshape(-1, 1))
    model = LinearRegression().fit(X_poly, y_train)

    y_pred_train = model.predict(X_poly)
    mse_train = mean_squared_error(y_train, y_pred_train)

    X_test_poly = poly.transform(X_test.reshape(-1, 1))
    y_pred_test = model.predict(X_test_poly)
    mse_test = mean_squared_error(y_test, y_pred_test)

    results.append((degree, mse_train, mse_test, model, poly))
    print(f"Степень {degree}: MSE(train)={mse_train:.6f}, MSE(test)={mse_test:.6f}")

best_degree, best_train_mse, best_test_mse, best_model, best_poly = min(
    results, key=lambda t: t[1]  # выбор по ошибке на обучающей выборке
)

print(f"\nОптимальная степень полинома: {best_degree}")
print(f"MSE(train)={best_train_mse:.6f}, MSE(test)={best_test_mse:.6f}")

x_plot = np.linspace(-2, 2, 400)
y_plot = runge_function(x_plot)
y_poly_plot = best_model.predict(best_poly.transform(x_plot.reshape(-1, 1)))

plt.figure(figsize=(10, 6))
plt.plot(x_plot, y_plot, label="Функция Рунге", color="blue")
plt.plot(x_plot, y_poly_plot, label=f"Оптимальный полином степени {best_degree}", color="red")
plt.scatter(X_train, y_train, color="black", marker="o", label="Обучающая выборка")
plt.scatter(X_test, y_test, color="green", marker="x", label="Контрольная выборка")
plt.title("Аппроксимация функции Рунге оптимальным полиномом")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```

Задание №2 (4 вариант):

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.

Функция по варианту:

$$f(x_1, x_2) = x_1^3 + x_2^2 - x_1 \cdot x_2 - 2x_1 + 3x_2 - 4$$

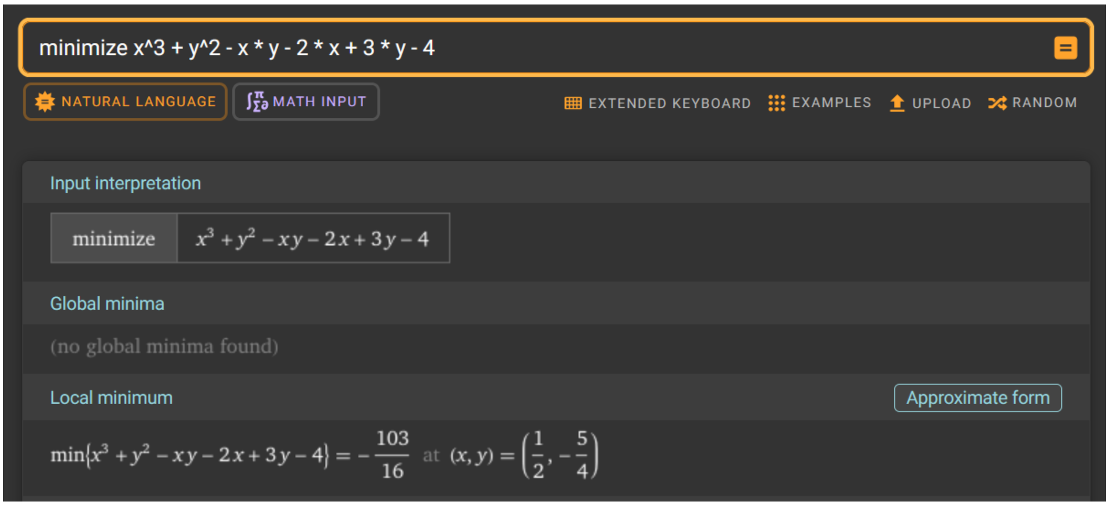


Рисунок 4 Координаты и значение функции в точке минимума

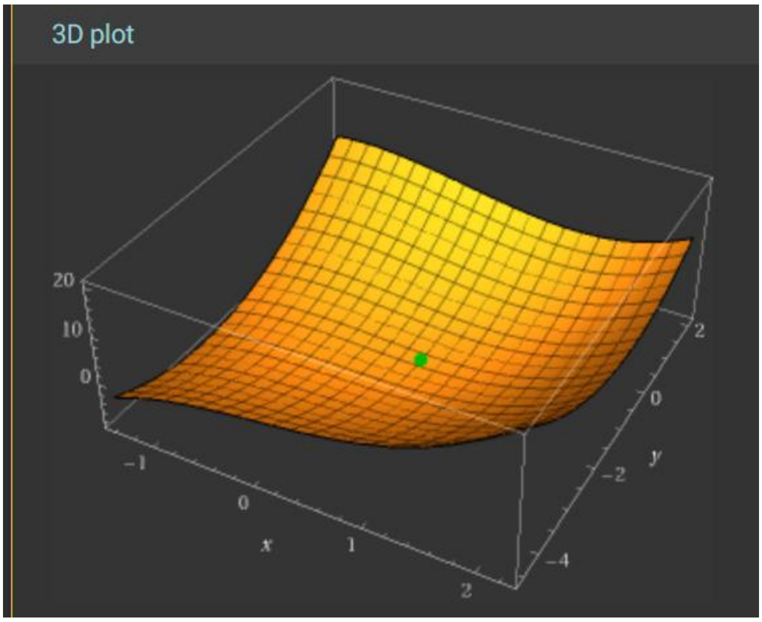


Рисунок 5 Построенный график

Вывод: в ходе лабораторной работы я построил полиномиальную модель для аппроксимации функции Рунге, подобрал её оптимальную степень и оценил качество на контрольной выборке. Выявлены признаки переобучения модели.