

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Лабораторная работа №1  
по дисциплине  
“Численные методы анализа ”  
Семестр II

Выполнили: студенты  
Таряник Антон Александрович  
гр. J3111  
ИСУ 467677  
Мавров Артём Николаевич  
гр. J3113  
ИСУ 466574  
Дубовик Никита Спартакович  
гр. J3110  
ИСУ 465809

Отчет сдан: 15.03.2025

Санкт-Петербург  
2025

# Цель

Предложить среднеквадратическую аппроксимацию табличной функции многих переменных, проанализировать чувствительность точного решения к ошибкам округления, проверить сходимость расчетных и исходных данных.

# Задачи

- Составить в матричном виде систему линейных алгебраических уравнений для поиска коэффициентов среднеквадратического приближения  $g$ , вычислить число обусловленности матрицы. Система составляется на основе выборки `train`.
- Найти решение системы при помощи обратной матрицы, проверить полученное приближение с использованием данных из обучающей и из тестовой выборки отдельно (рассчитать метрику  $\text{mean}_{absolute\ error, y}('MEDV')$ )

# Ход решения

## Импорт необходимых библиотек, датасета

```
1 from google.colab import drive
2 import numpy as np
3 import pandas as pd
4 from sklearn.linear_model import LinearRegression
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import StandardScaler, PolynomialFeatures
7 from sklearn.metrics import mean_absolute_error
8 import matplotlib.pyplot as plt
9 pd.set_option('display.width', 1000)
10
11 csv_file_path = '/content/drive/MyDrive/NMA_labs/housing.csv'
12 data = pd.read_csv(csv_file_path, encoding='cp1252')
```

## Задача 1

```
1 k = 13
2 X, y = data.drop([data.columns[k]], axis = 1), data[data.columns[k]]
3
4 poly = PolynomialFeatures(1, include_bias = False, interaction_only = True)
5
6 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 28)
7 X_train, X_test = poly.fit_transform(X_train), poly.fit_transform(X_test)
8 X_train.shape, X_test.shape
9
10 scale = StandardScaler(with_mean = False, with_std = False)
11
12 model = LinearRegression(fit_intercept = False)
13 model = model.fit(X_train, y_train)
14
15 mean_absolute_error(model.predict(X_train), y_train), mean_absolute_error(model.predict(X_test),
```

Выбираем как цель столбец 14, неизвестные с 1 по 13. Конструируем признаки, делим на трейн и тест. Далее обучается линейная модель и считается средняя абсолютная ошибка при тестировании, коэффициент детерминации, число ненулевых.

## Тестирование задачи 1

```
1 poly_features = PolynomialFeatures(degree=2)
2
3 X_train = scale.fit_transform(X_train)
4 X_test = scale.transform(X_test)
5 X_train = poly_features.fit_transform(X_train)
6 X_test = poly_features.transform(X_test)
7
8 a = np.linalg.pinv(X_train.T @ X_train) @ X_train.T @ y_train
9 mean_absolute_error(X_train@a, y_train), mean_absolute_error(X_test@a, y_test)
```

Повторение расчета для матриц  $X_{\text{train}}$ ,  $X_{\text{test}}$ ,  $PolynomialFeatures$ .

## Задача 2

```
1 X_train_with_bias = np.hstack((np.ones((X_train.shape[0], 1)), X_train))
2 X_test_with_bias = np.hstack((np.ones((X_test.shape[0], 1)), X_test))
3
4 A_train = X_train_with_bias.T @ X_train_with_bias
5 b_train = X_train_with_bias.T @ y_train
6 condition_number = np.linalg.cond(A_train)
7 print(f"Condition number: {condition_number:.2e}")
8
9 A_inv_train = np.linalg.pinv(A_train)
10
11 c_train = np.dot(A_inv_train, b_train)
12
13 y_pred_train = X_train_with_bias @ c_train
14 y_pred_test = X_test_with_bias @ c_train
15
16 mae_train = mean_absolute_error(y_train, y_pred_train)
17 mae_test = mean_absolute_error(y_test, y_pred_test)
18
19 print(f"MAE on the training sample: {mae_train}")
20 print(f"MAE on the testing sample: {mae_test}")
21 }
```

Составляется матрица методом наименьших квадратов, транспонирование матрицы и решение для обучающей выборки. Далее рассчитывается метрика для обучающей и средней выборки.

## Визуализация результатов

```
1 plt.figure(figsize=(10, 5))
2
3 plt.subplot(1, 2, 1)
4 plt.scatter(y_train, y_pred_train)
5 plt.plot([y_train.min(), y_train.max()], [y_train.min(), y_train.max()], 'k--')
6 plt.xlabel('Actual values')
7 plt.ylabel('Predicted values')
8 plt.title('Training sample')
9
10 plt.subplot(1, 2, 2)
11 plt.scatter(y_test, y_pred_test)
12 plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--')
13 plt.xlabel('Actual values')
```

```
14 plt.ylabel('Predicted values')
15 plt.title('Test sample')
16
17 plt.tight_layout()
18 plt.show()
19 }
```

## Вывод

В ходе выполнения лабораторной работы была построена система линейных алгебраических уравнений для среднеквадратического приближения табличной функции. Были проведены расчеты для обучающей и тестовой выборки, а также вычислено число обусловленности матрицы.

Полученные результаты подтверждают, что число обусловленности матрицы оказывает значительное влияние на точность аппроксимации. Использование полиномиальных признаков и масштабирования данных может как улучшить, так и ухудшить точность предсказаний, в зависимости от параметров модели. На основе расчетов сделан вывод о целесообразности выбора параметров модели с учетом их влияния на ошибки предсказания.

## Приложения

Ссылка на Google colab с лабораторной работой.