Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет программной инженерии и компьютерной техники*

Системы искусственного интеллекта

Лабораторная работа №3

Линейная регрессия

Группа: P3324

Выполнил: Маликов Глеб Игоревич

Преподаватель:

Королёва Юлия Александровна

Санкт-Петербург

2024г.

**Оглавление**

[Введение 3](#_Toc178326491)

[Описание метода 4](#_Toc178326492)

[Реализация метода 5](#_Toc178326493)

[Результаты выполнения 6](#_Toc178326494)

[Визуализация датасета 6](#_Toc178326495)

[Обработка данных 8](#_Toc178326496)

[Результаты моделей 8](#_Toc178326497)

[Модель 1 9](#_Toc178326498)

[Модель 2 9](#_Toc178326499)

[Модель 3 9](#_Toc178326500)

[Бонусная модель 10](#_Toc178326501)

[Анализ и сравнение результатов 10](#_Toc178326502)

[Примеры использования метода 11](#_Toc178326503)

# Введение

В данной лабораторной работе целью является реализация метода линейной регрессии для анализа данных об успеваемости студентов. Лабораторная работа включает подготовку данных, их предобработку, разделение на обучающие и тестовые выборки, а также обучение нескольких моделей линейной регрессии с различными наборами признаков. Кроме того, в качестве бонуса был введены синтетические признаки, чтобы улучшить качество модели.

# Описание метода

Метод линейной регрессии используется для моделирования зависимости между одной зависимой переменной (в данной работе — индекс успеваемости студентов) и одной или несколькими независимыми переменными (время подготовки, предыдущие оценки и т.д.). Принцип работы метода основан на нахождении такой прямой линии (гиперплоскости в случае многомерных данных), которая минимизирует сумму квадратов отклонений предсказанных значений от фактических. Для этого используется метод наименьших квадратов.

Линейная регрессия представляет собой метод нахождения линейной зависимости между зависимой переменной y и набором независимых переменных X. Модель можно выразить как:

где y — предсказанное значение, θ0 — свободный член, а θ1,…,θn— коэффициенты, которые необходимо оптимизировать.

Алгоритм градиентного спуска применяется для минимизации функции стоимости J(θ) которая является среднеквадратичной ошибкой:

где m — количество обучающих примеров, y^i — предсказанное значение, а yi — фактическое значение.

Процесс обновления коэффициентов на каждом шаге выглядит следующим образом:

где α — шаг обучения (learning rate).

# Реализация метода

Основные этапы реализации:

1. Считывание и визуализация данных, расчет основных статистических показателей (среднее, стандартное отклонение и т.д.).
2. Предобработка данных: удаление строк с пропущенными значениями, кодирование категориальных признаков и нормировка.
3. Разделение данных на обучающую и тестовую выборки в соотношении 80:20.
4. Реализация линейной регрессии с использованием метода наименьших квадратов
5. Обучение трех моделей с различными наборами признаков, а также бонусной модели с добавлением синтетических признаков. Для оценки качества моделей использовался коэффициент детерминации (R²).

Ключевой код для градиентного спуска:

def gradient\_descent(X, y, theta, learning\_rate=0.01, num\_iterations=1000):  
 assert X.shape[0] == y.shape[0], "X and y must have the same number of rows"  
 assert X.shape[1] == theta.shape[0], "X and theta must have the same number of columns"  
 print(f"Training {num\_iterations} iterations with learning rate {learning\_rate}")  
 m = len(y)  
 for i in range(num\_iterations):  
 predictions = X.dot(theta)  
 gradients = (1 / m) \* X.T.dot(predictions - y) # compute gradient  
 theta = theta - learning\_rate \* gradients # update theta  
   
 if i % 100 == 0 or i == num\_iterations - 1:  
 cost = (1 / (2 \* m)) \* np.sum((predictions - y) \*\* 2)  
 print(f"Iteration {i}: Cost {cost}")  
   
 print(f"Training complete.")  
 return theta  
  
# Initialize theta (all zeros)  
theta = np.zeros(X\_train.shape[1])  
  
# Perform gradient descent to find the optimal theta  
theta = gradient\_descent(X\_train, y\_train, theta)

# Результаты выполнения

## Визуализация датасета

Датасет имеет признаки “Hours Studied”, “Previous Scores”, “Sleep Hours”, “Sample Question Practiced” и “Performance Index” который необходимо предсказывать.

Ниже приведена визуализация данных датасета:

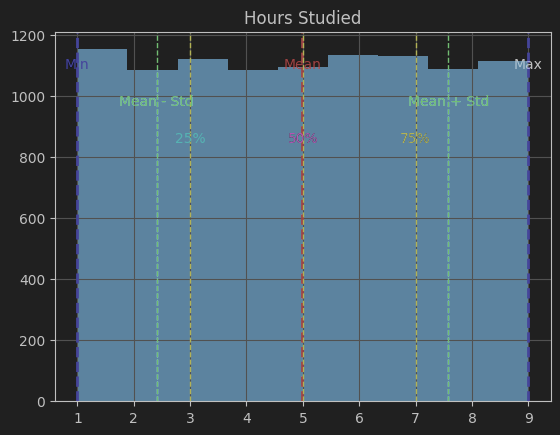


Рисунок 1 - Статистика "Hours Studied"

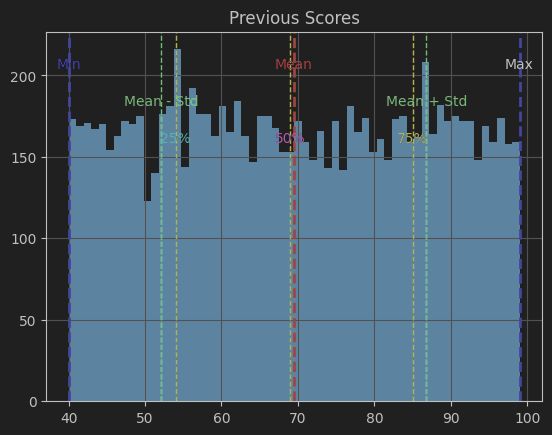
\

Рисунок 2 - Статистика "Previous Scores"

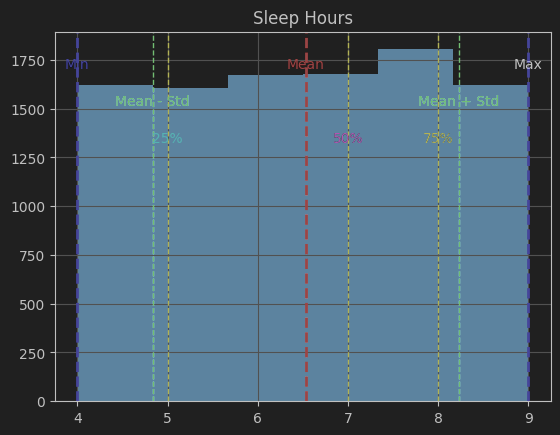


Рисунок 3 - Статистика "Sleep Hours"

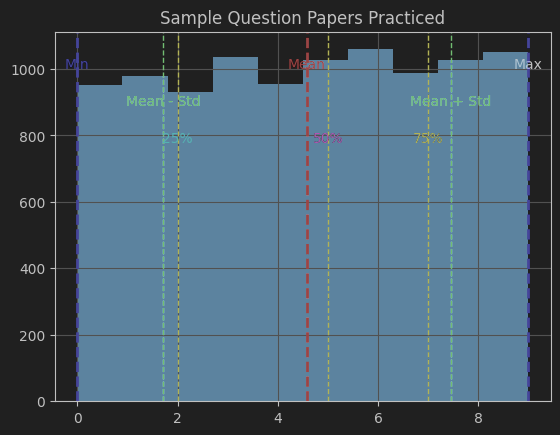


Рисунок 4 - Статистика "Sample Question Papers Practiced"

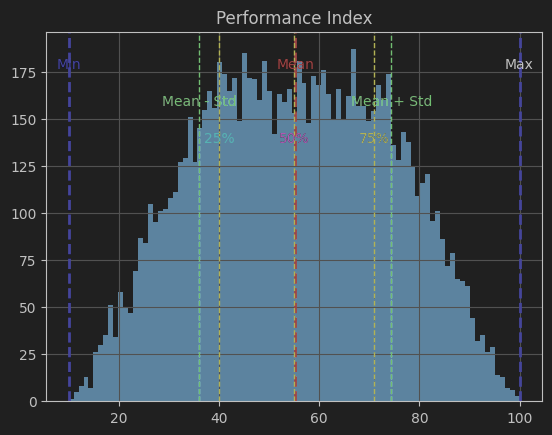


Рисунок 5 - Статистика "Performance Index"

## Обработка данных

Были удалены строки с пропущенными значениями:

Missing values per column:

Hours Studied 0

Previous Scores 0

Extracurricular Activities 0

Sleep Hours 0

Sample Question Papers Practiced 0

Performance Index 0

dtype: int64

Deleted rows: 0

Категориальный признак “Extracurricular Activities” был закодирован в значения 0 и 1:

data['Extracurricular Activities'] = data['Extracurricular Activities'].map({'Yes': 1, 'No': 0})

Была произведена нормализация данных по формуле:

## Результаты моделей

Были построены 4 модели со следующими признаками:

1. Модель 1: все признаки
2. Модель 2: только признаки 'Hours Studied', 'Hours Slept'
3. Модель 3: только признаки 'Previous Scores', 'Extracurricular Activities', 'Sample Question Papers Practiced'
4. Бонусная модель: все признаки с добавлением синтетических признаков 'Preparation', который равен произведению признаков 'Hours Studied' и 'Sample Question Papers Practiced'  
   и признак 'Time Left', равен разности 24 часов и признаков 'Sleep Hours' и 'Hours Studied'

Все модели тренировались по функции “gradient\_descent” описанному выше, с параметрами 0.01 learning rate и 1000 итерации.

Для оценки моделей используется функция для вычисления коэффициента детерминации

где RSS(Residual Sum of Squares) — сумма квадратов остатков, или невязок между фактическими и предсказанными значениями, TSS (Total Sum of Squares) — общая сумма квадратов отклонений фактических значений от их среднего.

### Модель 1

Iteration 0: Cost 0.5012525323570195

Iteration 100: Cost 0.07250165481554842

Iteration 200: Cost 0.014642394352336344

Iteration 300: Cost 0.0068184674457541835

Iteration 400: Cost 0.005758330655834117

Iteration 500: Cost 0.005614390506903165

Iteration 600: Cost 0.005594807508577631

Iteration 700: Cost 0.0055921379186112145

Iteration 800: Cost 0.005591773277351132

Iteration 900: Cost 0.005591723374236819

Iteration 999: Cost 0.0055917165536208645

Коэффициент детерминации 0.9883697440959182

## Модель 2

Iteration 0: Cost 0.5012525323570195

Iteration 100: Cost 0.4389155882389505

Iteration 200: Cost 0.4305504571328264

Iteration 300: Cost 0.4294277880437181

Iteration 400: Cost 0.4292770988311313

Iteration 500: Cost 0.4292568703370281

Iteration 600: Cost 0.4292541545511125

Iteration 700: Cost 0.4292537898998782

Iteration 800: Cost 0.4292537409321899

Iteration 900: Cost 0.4292537343557501

Iteration 999: Cost 0.4292537334752035

Коэффициент детерминации 0.13463663177021534

## Модель 3

Iteration 0: Cost 0.5012525323570195

Iteration 100: Cost 0.13633279638736773

Iteration 200: Cost 0.08768570899498425

Iteration 300: Cost 0.08119840779849144

Iteration 400: Cost 0.08033300968298619

Iteration 500: Cost 0.08021752897828406

Iteration 600: Cost 0.08020211405249353

Iteration 700: Cost 0.08020005575129477

Iteration 800: Cost 0.08019978082970437

Iteration 900: Cost 0.08019974409824579

Iteration 999: Cost 0.08019973920463319

Коэффициент детерминации 0.8355939872997946

## Бонусная модель

Iteration 0: Cost 0.5012525323570195

Iteration 100: Cost 0.06560733315559464

Iteration 200: Cost 0.013597194101308795

Iteration 300: Cost 0.006675955737193661

Iteration 400: Cost 0.005739118171591589

Iteration 500: Cost 0.005611795521143101

Iteration 600: Cost 0.005594449941518166

Iteration 700: Cost 0.0055920816479511385

Iteration 800: Cost 0.005591757571047027

Iteration 900: Cost 0.00559171312465679

Iteration 999: Cost 0.0055917070346454345

Коэффициент детерминации 0.988370426824619

## Анализ и сравнение результатов

Модель 1 показывает наилучшие результаты (R² = 0.9884), так как использует все признаки. Высокий коэффициент детерминации говорит о том, что данная модель объясняет почти всю изменчивость данных.

Модель 2 (используются только два признака) демонстрирует низкое качество (R² = 0.1346). Этот результат указывает на то, что данные признаки недостаточно объясняют целевую переменную.

Модель 3 с набором других признаков показывает лучшее качество, чем Модель 2, с коэффициентом детерминации R² = 0.8356. Это говорит о том, что такие признаки, как предыдущие оценки и практика решения задач, более значимы для предсказания результата.

Бонусная модель, включающая синтетические признаки, незначительно улучшает результаты первой модели, имея почти идентичные значения функции стоимости и коэффициента детерминации (R² = 0.9884).

# Примеры использования метода

Линейная регрессия часто применяется для предсказания зависимых переменных на основе известных факторов. Например, в экономике метод используют для прогнозирования продаж в зависимости от таких факторов, как цена и затраты на маркетинг. Линейная зависимость между переменными позволяет оценить, как изменение одного фактора влияет на результат, что делает этот метод простым и понятным инструментом для анализа.

В медицине линейная регрессия применяется для анализа влияния различных факторов, таких как возраст, вес или количество потребляемых калорий, на показатели здоровья, например, кровяное давление. Этот метод удобен в ситуациях, где есть предположение о линейной зависимости между признаками, и важна интерпретируемость результатов.