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 №6

Логистическая регрессия

Группа: P3324

Выполнил: Маликов Глеб Игоревич

Преподаватель:

Королёва Юлия Александровна

Санкт-Петербург

2024г.

**Оглавление**

[Введение 3](#_Toc180234401)

[Описание метода 4](#_Toc180234402)

[Реализация метода 5](#_Toc180234403)

[Результаты выполнения 6](#_Toc180234404)

[Примеры использования метода 11](#_Toc180234405)

# Введение

Целью данной лабораторной работы является реализация и исследование алгоритма логистической регрессии для задачи бинарной классификации на примере датасета, содержащего информацию о пациентах с диабетом. Основные задачи включают предварительную обработку данных, визуализацию статистики, разделение данных на обучающую и тестовую выборки, реализацию метода логистической регрессии "с нуля" без использования специализированных библиотек, исследование влияния гиперпараметров на производительность модели, а также оценку модели с использованием различных метрик качества.

# Описание метода

Логистическая регрессия — это статистический метод, используемый для моделирования вероятности принадлежности объекта к одному из двух классов. В отличие от линейной регрессии, которая предсказывает непрерывные значения, логистическая регрессия предсказывает вероятность принадлежности к классу, используя функцию активации сигмоиды. Основным назначением метода является бинарная классификация, однако его можно расширить и для многоклассовых задач. Принцип работы заключается в нахождении оптимальных весовых коэффициентов, минимизирующих функцию потерь (логарифмическая потеря), с помощью методов оптимизации, таких как градиентный спуск или метод Ньютона.

# Реализация метода

В реализации логистической регрессии использовались библиотеки NumPy и Pandas. Основные компоненты алгоритма включают:

* Функция сигмоиды: преобразует линейную комбинацию признаков в вероятность принадлежности к классу.

def sigmoid(self, z):  
 return 1 / (1 + np.exp(-z))

* Функция потерь (логарифмическая потеря): оценивает расхождение между предсказанными вероятностями и истинными метками классов.

def log\_loss(self, h, y):  
 # Добавляем небольшое значение для предотвращения логарифма от 0  
 epsilon = 1e-15  
 h = np.clip(h, epsilon, 1 - epsilon)  
 return (-y \* np.log(h) - (1 - y) \* np.log(1 - h)).mean()

* Методы оптимизации: Реализованы градиентный спуск, метод Ньютона и метод сопряженных градиентов для обновления весов модели.

def fit\_gradient\_descent(self, X, y):  
 for i in range(self.num\_iterations):  
 z = np.dot(X, self.theta)  
 h = self.sigmoid(z)  
 gradient = np.dot(X.T, (h - y)) / y.size  
 self.theta -= self.learning\_rate \* gradient  
   
 if i % 100 == 0:  
 loss = self.log\_loss(h, y)  
 print(f'Итерация {i}: loss {loss}')  
  
def fit\_newton(self, X, y):  
 for i in range(self.num\_iterations):  
 z = np.dot(X, self.theta)  
 h = self.sigmoid(z)  
 gradient = np.dot(X.T, (h - y)) / y.size  
 # Гессиан  
 diag = h \* (1 - h)  
 H = np.dot(X.T, X \* diag[:, np.newaxis]) / y.size  
   
 # Обновление весов  
 try:  
 delta = np.linalg.inv(H).dot(gradient)  
 except np.linalg.LinAlgError:  
 print("Гессиан вырожден. Прекращаем обучение.")  
 break  
   
 self.theta -= delta  
   
 if i % 100 == 0:  
 loss = self.log\_loss(h, y)  
 print(f'Итерация {i}: loss {loss}')

# Результаты выполнения

Для обработки данных были удалены строки с нулевыми значениями для столбцов Glucose, BloodPressure, SkinThickness, BMI, так как они невозможны. Далее данные были нормализированы с помощью мин-макс нормализации.

Ниже приведены статистические данные столбцов:

A screen shot of a graph

Description automatically generated

Рисунок 1 - Статистика беременности

A graph of blood sugar levels

Description automatically generated with medium confidence

Рисунок 2 - Статистика глюкозы

A screen shot of a graph

Description automatically generated

Рисунок 3 - Статистика давления

A graph of a graph

Description automatically generated with medium confidence

Рисунок 4 - Статистика толщины кожи

A screenshot of a graph

Description automatically generated

Рисунок 5 - Статистика инсулина

A screen shot of a graph

Description automatically generated

Рисунок 6 - Статистика BMI

A screen shot of a graph

Description automatically generated

Рисунок 7 - Статистика Pedigree

A graph of a number of people

Description automatically generated with medium confidence

Рисунок 8 - Статистика возраста

Модели использовали 80% данных как тестовые. Были использованы следующие гиперпараметры:

learning\_rates = [1, 0.5, 0.1, 0.01, 0.001]  
num\_iterations = [10, 100, 1000, 10000]  
optimization\_methods = ['gradient\_descent', 'newton']

Ниже показана таблица метрик лучших 10 моделей по метрике f1 score.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метод | Коэф. обучения | Итерации | accuracy | precision | recall | f1\_score |
| gradient\_descent | 1 | 1000 | 0.821 | 0.733 | 0.667 | 0.698 |
| gradient\_descent | 0.5 | 1000 | 0.821 | 0.733 | 0.667 | 0.698 |
| gradient\_descent | 0.1 | 10000 | 0.821 | 0.733 | 0.667 | 0.698 |
| gradient\_descent | 1 | 100 | 0.830 | 0.800 | 0.606 | 0.690 |
| gradient\_descent | 0.01 | 10000 | 0.830 | 0.800 | 0.606 | 0.690 |
| gradient\_descent | 0.1 | 1000 | 0.830 | 0.800 | 0.606 | 0.690 |
| newton | 1 | 100 | 0.811 | 0.724 | 0.636 | 0.677 |
| newton | 1 | 10 | 0.811 | 0.724 | 0.636 | 0.677 |
| newton | 0.001 | 1000 | 0.811 | 0.724 | 0.636 | 0.677 |
| newton | 1 | 10000 | 0.811 | 0.724 | 0.636 | 0.677 |

Таблица 1 - Результаты лучших 10 моделей

Для визуализации результатов также сделаны следующие графики.

A graph of different colored bars

Description automatically generated

Рисунок 9 - Результаты моделей по Accuracy

A screenshot of a graph

Description automatically generated

Рисунок 10 - Результаты моделей по Precision

A graph of different colored bars

Description automatically generated with medium confidence

Рисунок 11 - Результаты моделей по Recall

A screenshot of a graph

Description automatically generated

Рисунок 12 - Результаты моделей по F1-Score

Лучшая модель была найдена по сортировке F1-Score, а затем по Accuracy. Ниже приведены её результаты.

Метод оптимизации: gradient\_descent

Коэффициент обучения: 1.0

Количество итераций: 1000

Accuracy: 0.8208

Precision: 0.7333

Recall: 0.6667

F1-Score: 0.6984

# Примеры использования метода

Логистическая регрессия широко применяется в различных областях, где требуется бинарная классификация. Примеры использования включают:

* Медицина: Прогнозирование наличия или отсутствия заболевания у пациента на основе клинических показателей. В данной лабораторной работе логистическая регрессия использовалась для классификации пациентов с диабетом.
* Финансы: Оценка риска дефолта заемщика по кредиту, что помогает банкам принимать обоснованные решения о выдаче кредитов.
* Социальные науки: Анализ факторов, влияющих на выбор избирателей, и прогнозирование результатов выборов.

Метод логистической регрессии выбран в данной работе благодаря его интерпретируемости, эффективности при работе с линейными зависимостями и способности оценивать вероятности принадлежности к классам, что является важным для принятия решений на основе модели.