Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа №4

по курсу «Моделирование»

«Модели машинного обучения (логистическая регрессия)»

Студент группы ИУ9-81Б Окутин Д. А.

Преподаватель Домрачева А. Б.

Москва 2025

Цель работы

Цель лабораторной работы — изучить и применить модель логистической регрессии для анализа данных, а также оценить её эффективность с использованием различных метрик качества.

Постановка задачи

Имеется небольшая база данных больных диабетом I, II типа. Необходимо для выданного датасета:

1. Построить диаграмму рассеяния
2. Оценить совместное распределение для множества величин
3. Реализовать самостоятельно алгоритм логистической регрессии
4. Оценить её точность на основе F-меры, сравнить с библиотечной реализацией этой модели на Python.

Теоретические сведения

Диаграмма рассеяния — способ визуализации взаимосвязи между двумя (или более) количественными переменными. Для двух случайных величин X и Y, каждая точка на диаграмме представляет пару наблюдаемых значений (x_i, y_i).

- Горизонтальная ось (ось абсцисс) отображает значения переменной X;
- Вертикальная ось (ось ординат) отображает значения переменной Y;
- Каждая точка (x_i, y_i) на графике представляет одно наблюдение.

Корреляция — статистическая мера, отражающая степень линейной зависимости между двумя переменными. Взаимозависимость двух величин называется взаимная корреляция. График корреляции позволяет утверждать, что две величины явно взаимосвязаны, или взаимно коррелируют. В том случае, если

существует прямая зависимость величин (чем больше одна, тем больше другая), то говорят, что корреляция положительная. В том случае, если существует обратная зависимость величин (чем больше одна, тем меньше другая), то говорят, что корреляция отрицательная. В предположении, что целевой показатель связан с другими показателями линейно, взаимосвязь оценивается с помощью коэффициента корреляции Пирсона.

Коэффициент корреляции Пирсона Для количественной оценки используется коэффициент корреляции Пирсона (r):

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}},$$

где:

- $\bar{x}, \bar{y}$ — средние значения переменных,
- n — количество наблюдений.

Интерпретация:

- $r = 1$: Полная прямая линейная зависимость
- $r = -1$: Полная обратная линейная зависимость
- $r = 0$: Отсутствие линейной связи
- $|r| > 0.7$: Сильная корреляция
- $|r| < 0.3$: Слабая корреляция

Корреляционная матрица - таблица, где каждый элемент r_{ij} показывает корреляцию между i-м и j-м признаками:

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}$$

Свойства:

- Диагональные элементы равны 1 ($r_{ii} = 1$)
- Симметричность ($r_{ij} = r_{ji}$)
- Визуализируется тепловой картой для анализа паттернов

Ограничения корреляции Пирсона:

- Отражает только линейные зависимости
- Чувствительна к выбросам
- Не показывает причинно-следственные связи

Логистическая регрессия — метод классификации, используемый для предсказания вероятности принадлежности объекта к одному из двух классов. В отличие от линейной регрессии, где выходная переменная непрерывна, здесь применяется **сигмоидная функция** для преобразования линейной комбинации признаков в диапазон [0, 1].

После построения логистической регрессии необходимо оценить её эффективность. Для бинарной классификации используются четыре базовых категории предсказаний:

- **True Positive (TP)** — верно предсказанные положительные классы;
- **True Negative (TN)** — верно предсказанные отрицательные классы;
- **False Positive (FP)** — ложные срабатывания (ошибочно положительные);
- **False Negative (FN)** — пропущенные цели (ошибочно отрицательные).

На основе матрицы ошибок рассчитываются следующие показатели:

- **Точность (Precision)**: Доля релевантных результатов среди всех положительных предсказаний

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Полнота (Recall)**: Доля найденных релевантных результатов

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-мера**: Баланс между точностью и полнотой

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Accuracy**: Общая доля верных предсказаний

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Интерпретация метрик:

- **Precision** критичен в задачах, где ложные срабатывания дорого обходятся (например, спам-фильтры)
- **Recall** важен в медицинской диагностике, где пропуск заболевания недопустим
- **F1** используют при дисбалансе классов или необходимости компромисса между точностью и полнотой

Практическая реализация

В фргаменте кода в листинге 1 подготавливаем данные для обучения модели, делим на обучающую/тестовую выборки, готовим структуру для обучения и тестирования.

Листинг 1: Подготовка данных

```
df = pd.read_csv('diabetes.csv')

X = df.drop(columns=['Outcome'], axis=1)
y = df['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)
```

В фрагменте кода, представленном в листинге 2 строится диаграмма рассеяния с помощью метода pairplot из библиотеки seaborn.

Листинг 2: Построение диаграммы рассеивания и матрицы корреляции

```
sns.pairplot(df,diag_kind='kde')
plt.show()

plt.figure (figsize = [8,6],  dpi = 130 )
plt.title ("Correlation Graph" , fontsize = 11 )
sns.heatmap (df.corr(), annot = True , cmap="coolwarm" )
```

В фрагменте кода в листинге 3 представлена собственная реализация логистической регрессии в классе CustomLogisticRegression, который включает методы: fit — для обучения модели с использованием градиентного спуска и predict — для предсказания классов на основе обученной модели.

Листинг 3: Реализация логистической регрессии

```
class CustomLogisticRegression:
    def __init__(self, learning_rate=0.01, n_iters=1000):
        self.lr = learning_rate
        self.n_iters = n_iters
        self.weights = None
        self.bias = None

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

    def fit(self, X, y):
        n_samples, n_features = X.shape

        self.weights = np.zeros(n_features)
        self.bias = 0

        for _ in range(self.n_iters):
            linear_model = np.dot(X, self.weights) + self.bias
            y_pred = self._sigmoid(linear_model)

            dw = (1/n_samples) * np.dot(X.T, (y_pred - y))
            db = (1/n_samples) * np.sum(y_pred - y)

            self.weights -= self.lr * dw
            self.bias -= self.lr * db

    def predict_proba(self, X):
```

```
        linear_model = np.dot(X, self.weights) + self.bias
        return self._sigmoid(linear_model)

    def predict(self, X, threshold=0.5):
        probabilities = self.predict_proba(X)
        return (probabilities >= threshold).astype(int)
```

В фрагменте кода, представленном в листинге 4, рассчитываются метрики для оценки качества моделей.

Листинг 4: Вычисление основных метрик

```
def calculate_metrics(y_true, y_pred):
    y_true = np.array(y_true)
    y_pred = np.array(y_pred)

    if y_true.shape != y_pred.shape:
        raise ValueError("                y_true    y_pred
                         ")

    TP = np.sum((y_true == 1) & (y_pred == 1))
    TN = np.sum((y_true == 0) & (y_pred == 0))
    FP = np.sum((y_true == 0) & (y_pred == 1))
    FN = np.sum((y_true == 1) & (y_pred == 0))

    accuracy = (TP + TN) / (TP + TN + FP + FN)
    precision = TP / (TP + FP) if (TP + FP) > 0 else 0.0
    recall = TP / (TP + FN) if (TP + FN) > 0 else 0.0

    return {
        'accuracy': round(accuracy, 4),
        'precision': round(precision, 4),
        'recall': round(recall, 4),
        'f1': round(2*precision*recall/(precision+recall),4),
        'confusion_matrix': {
            'TP': TP,
            'TN': TN,
            'FP': FP,
            'FN': FN
        }
    }
```

Результаты работы

На рисунке ниже показана диаграмма рассеяния характеристик, описанных во входном наборе данных.

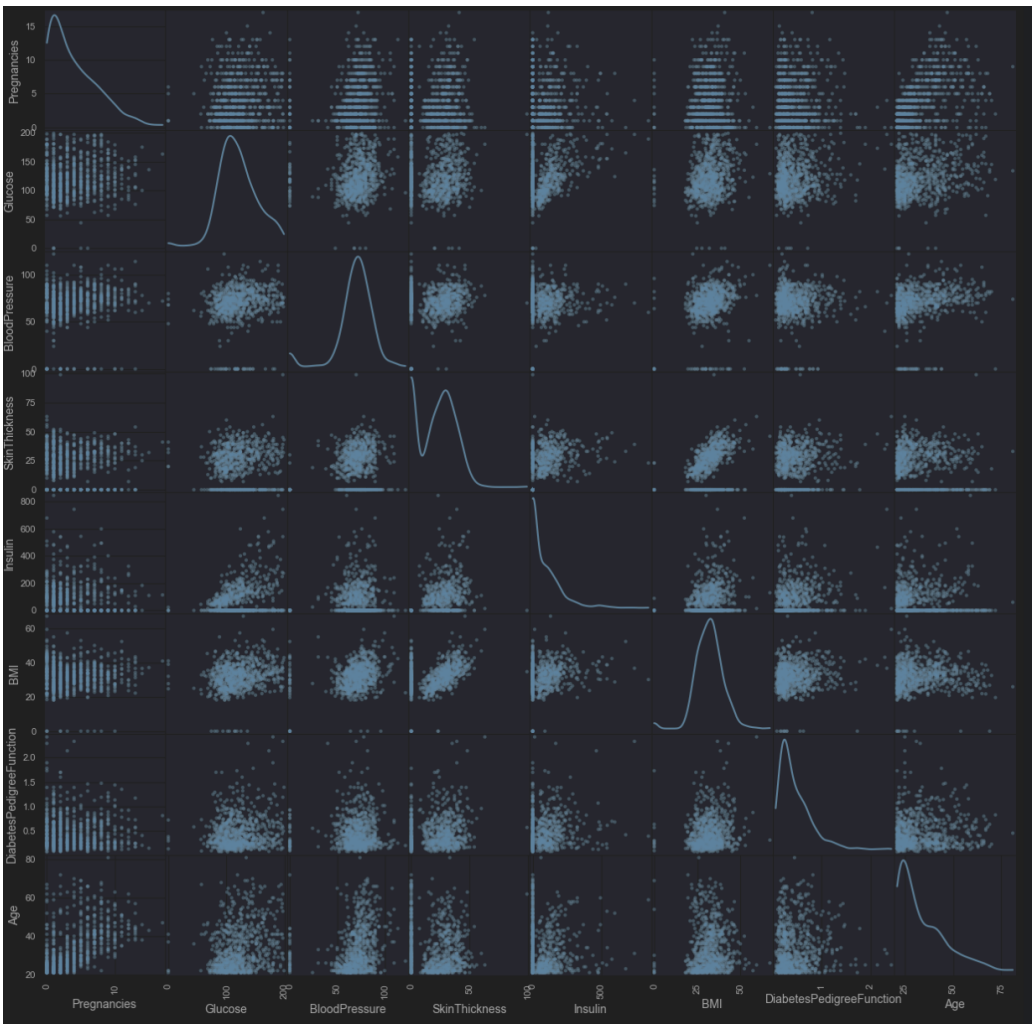

Рис. 1 — Диаграмма рассеяния

Матрица корреляции признаков в представленном наборе данных представлена на рисунке 2.

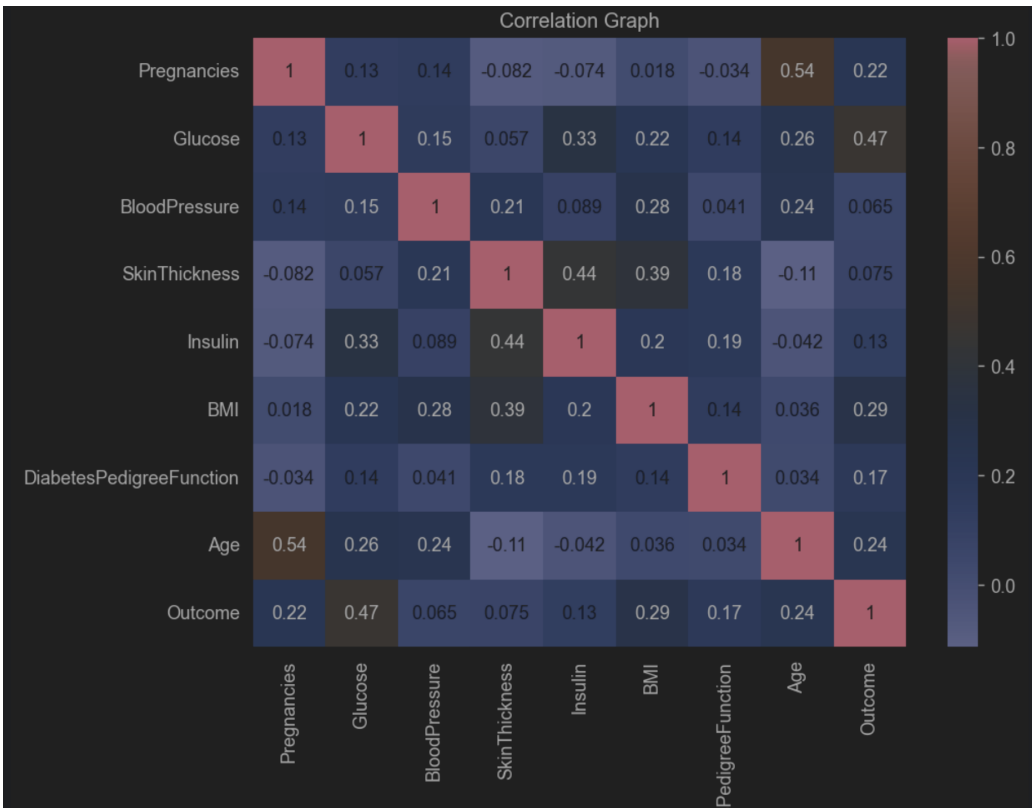

Рис. 2 — Матрица корреляции

На основе представленной матрицы корреляции можно сделать следующие выводы:

- **Ключевы признаки для прогнозирования параметра Outcome**:
  - **Glucose (0.47)**: Положительная корреляция. Уровень глюкозы — основной предиктор.
  - **BMI (0.29)**: Умеренная связь. Ожирение повышает риск диабета.
  - **Age (0.24)**: Слабая корреляция. Возраст частично влияет на вероятность заболевания.
- **Взаимосвязи между признаками**:
  - **Insulin ↔ SkinThickness (0.44)**: Высокий уровень инсулина связан с увеличением толщины кожи.
  - **Glucose ↔ Insulin (0.33)**: Умеренная связь уровня глюкозы и инсулина.

Таблица 1: Метрики качества моделей

Модель	Accuracy	Precision	Recall	F1-score
My Logistic Regression	0.6883	0.5467	0.7455	0.63
Logistic Regression	0.7468	0.6379	0.6727	0.65

На основании полученных метрик качества моделей логистической регрессии можно сделать следующие выводы о свойствах и структуре исходного датасета:

- **Высокое значение Recall (до 0.7455)** указывает на то, что модель успешно выявляет большинство положительных примеров (Outcome = 1). Это может говорить о наличии информативных признаков, хорошо отделяющих класс 1.
- **Низкий Precision (около 0.55–0.64)** по сравнению с Recall может свидетельствовать о наличии классового дисбаланса — класс 1 встречается реже, чем класс 0. Также возможен шум в данных, из-за чего модель часто ошибочно классифицирует примеры как положительные.
- **Умеренные значения Accuracy (менее 0.75)** могут говорить о сложности задачи: либо границы между классами недостаточно чёткие, либо данные содержат выбросы, пропуски или неинформативные признаки.
- **Средний F1-score (0.63–0.65)** подтверждает, что модели не удаётся одновременно достичь высокой точности и полноты. Это может быть следствием как несбалансированности классов, так и наличия перекрывающихся признаков.

Можно сделать выводы о некоторых проблемах в данных:

- Классовый дисбаланс (преобладание одного из классов);
- Наличие шумных, нерелевантных или коррелирующих признаков;
- Сложная структура разделения классов;
- Ограниченный объём данных.

Заключение

В ходе работы получены навыки визуализации данных для анализа и самостоятельной реализации алгоритма логистической регрессии. Визуализация данных помогла лучше погрузиться в контекст задачи и выявить наиболее важные признаки, на которых стоит опираться при проработке модели.

Для моделей логистической регрессии сравнение метрик показало, что библиотечная модель превосходит ручную реализацию по всем измеренным показателям качества, что подчеркивает важность оптимизации гиперпараметров.

Также метрики явно указывают на то, что исходные данные не идеальны и требуют более тщательной предобработки.