|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное автономное 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 | СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ |

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

|  |
| --- |
| ***Обработка датасета*** |
|  |
|  |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ5-61Б |  |  |  | К.А. Сомов |
|  | (группа) |  | (подпись, дата) |  | (И.О. Фамилия) |
| Руководитель НИР |  |  |  |  | Ю. Е. Гапанюк |
|  |  |  | (подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |  |

*2025 г.*

**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)**

|  |  |  |
| --- | --- | --- |
| Утверждаю | | |
|  | | |
| Заведующий кафедрой | | ИУ5 |
|  | | (индекс) |
|  | В.И. Терехов | |
|  | (И.О. Фамилия) | |
| (подпись) |  | |
|  | (дата) | |

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| по теме | | Обработка датасета | | | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | |
| Студент группы | | | | ИУ5-61Б | | | |  | | | | | | | |
|  | | | | Сомов Кирилл Анатольевич | | | | | | | | | | | |
| Направленность НИР (учебная, исследовательская, практическая, производственная, др.) | | | | | | | | | | | | | | | |
| ИССЛЕДОВАТЕЛЬСКАЯ | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| Источник тематики (кафедра, предприятие, НИР) | | | | | | | | | | КАФЕДРА | | | | | |
| График выполнения НИР: | | | | | |  | | | | | | | | | |
| 25% к |  | | нед., 50% к | | |  | нед., 75% к | |  | | | нед., 75% к |  | нед |
| ***Техническое задание:*** | | | | |  | | | | | | | | | | |
| Обработка датасета | | | | | | | | | | | | | | | |
| Типовое исследование | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| ***Оформление научно-исследовательской работы:*** | | | | | | | | | | |  | | | | |
| Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4. | | | | | | | | | | | | | | | |
| Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.) | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |

Дата выдачи задания «07» \_февраля\_\_\_ 2025 г.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель НИР** |  |  |  | Ю.Е. Гапанюк |
|  |  | (подпись, дата) |  | (И.О. Фамилия) |
| **Студент** |  |  |  | К.А. Сомов |
|  |  | (подпись, дата) |  | (И.О. Фамилия) |

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Введение

Актуальность исследования

Заболевание диабетом — одна из самых серьезных проблем общественного здравоохранения в мире. Раннее выявление риска диабета позволяет своевременно принять профилактические меры, снизить вероятность осложнений и уменьшить нагрузку на медицинскую систему. Традиционная диагностика требует очных обследований и анализа множества параметров, что может быть затруднительно для широкого скрининга. Современные методы машинного обучения позволяют автоматизировать процесс прогнозирования на основе стандартных медицинских показателей, делая диагностику более доступной и быстрой.

Цель исследования

Разработать систему прогнозирования риска диабета с использованием моделей машинного обучения. Для достижения этой цели в работе выполнены следующие задачи:

Проведена предобработка и анализ данных (замена пропущенных значений, стандартизация признаков).

Обучены и протестированы несколько моделей: логистическая регрессия, случайный лес, градиентный бустинг.

Реализован удобный интерфейс на базе Streamlit, позволяющий интерактивно вводить медицинские параметры и получать вероятностный прогноз.

Объект исследования

Датасет Pima Indians Diabetes Database, содержащий информацию о 768 пациентах, включая такие медицинские параметры, как уровень глюкозы, артериальное давление, индекс массы тела, возраст и другие характеристики. Целевой признак — наличие или отсутствие диабета (бинарная классификация).

Практическая значимость

Разработанная система может быть использована в следующих целях:

Медицинская практика: как вспомогательный инструмент для предварительного скрининга пациентов.

Образование: для демонстрации применения алгоритмов машинного обучения в реальной задаче здравоохранения.

Прототипирование ИИ-приложений: как основа для расширения и внедрения в мобильные или веб-сервисы здравоохранения.

Проект охватывает все этапы типового ML-цикла: от загрузки и подготовки данных до обучения моделей, их сравнения и внедрения в виде работающего веб-приложения.

Постановка задачи

В рамках настоящего исследования была решена задача бинарной классификации — **прогнозирование наличия диабета** на основе медицинских показателей. Реализация включает полный цикл разработки модели машинного обучения, включая её демонстрацию через веб-интерфейс.

**Этапы работы включали:**

* **Поиск и выбор датасета.** В качестве источника данных выбран открытый датасет *Pima Indians Diabetes Dataset*, содержащий информацию о 768 женщинах индейского происхождения, включая 8 медицинских показателей и бинарный признак наличия диабета.
* **Разведочный анализ данных (EDA).** Произведён первичный анализ структуры данных. Были выявлены признаки с отсутствующими или некорректными значениями (например, нулевые значения глюкозы, давления, ИМТ и др.). Эти значения были заменены медианными значениями по признакам.
* **Предобработка данных.** Проведено масштабирование признаков с помощью StandardScaler. Все признаки являются числовыми, что упростило подготовку и позволило избежать этапа кодирования категориальных переменных.
* **Корреляционный анализ.** Вычислены корреляции между признаками и целевой переменной. Выявлены наиболее значимые факторы (глюкоза, ИМТ, возраст и др.), влияющие на вероятность заболевания.
* **Выбор метрик.** Для оценки качества моделей использовались следующие метрики:
  + **Accuracy** — как базовая метрика;
  + **Precision и Recall** — важны для оценки качества классификации при наличии возможного дисбаланса классов;
  + **ROC AUC** — для анализа способности моделей отделять классы.
* **Выбор и обучение моделей.** Были реализованы и обучены следующие модели:
  + **Random Forest** (ансамблевая модель);
  + **Gradient Boosting** (ансамблевая модель);
  + **Logistic Regression**;  
    *(возможно, будут добавлены ещё модели — например, KNN, SVM, если потребуется соответствие требованиям о 5 моделях).*
* **Формирование выборок.** Данные разделены на обучающую и тестовую выборки в соотношении 80/20 с использованием train\_test\_split.
* **Построение базового решения.** Все модели были обучены без подбора гиперпараметров, получены базовые значения метрик на тестовой выборке.
* **Подбор гиперпараметров.** Для моделей Random Forest и Gradient Boosting реализована возможность настройки следующих параметров через веб-интерфейс:
  + количество деревьев (n\_estimators);
  + максимальная глубина (max\_depth);
  + минимальное число объектов для разбиения (min\_samples\_split).  
    Настройка гиперпараметров производится интерактивно в веб-приложении, что позволяет пользователю наблюдать за изменением качества модели в реальном времени.
* **Повторная оценка качества.** После изменения параметров модели переобучаются, а новые метрики точности отображаются в интерфейсе для анализа улучшений по сравнению с baseline.
* **Визуализация и выводы.** Результаты сравнения моделей по точности выведены в интерфейсе приложения. Также отображается вероятность наличия диабета для введённых пользователем медицинских показателей.
* **Веб-приложение.** Построено с использованием фреймворка **Streamlit**. Пользователь может:
  + изменять гиперпараметры ансамблевых моделей;
  + вводить индивидуальные показатели здоровья;
  + получать прогноз вероятности наличия диабета от трёх моделей;
  + видеть оценку риска и соответствующую рекомендацию.

Последовательность действий

**1. Импорт библиотек**

Импортируем все необходимые библиотеки для анализа данных, визуализации и машинного обучения.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

**2. Загрузка и подготовка данных**

Загружаем датасет диабета, заменяем нули на медианные значения и масштабируем признаки.

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"

columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',

'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

df = pd.read\_csv(url, names=columns)

cols\_to\_fix = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']

df[cols\_to\_fix] = df[cols\_to\_fix].replace(0, np.nan)

df.fillna(df.median(), inplace=True)

**3. Первичный анализ данных**

Выводим общую информацию и статистики.

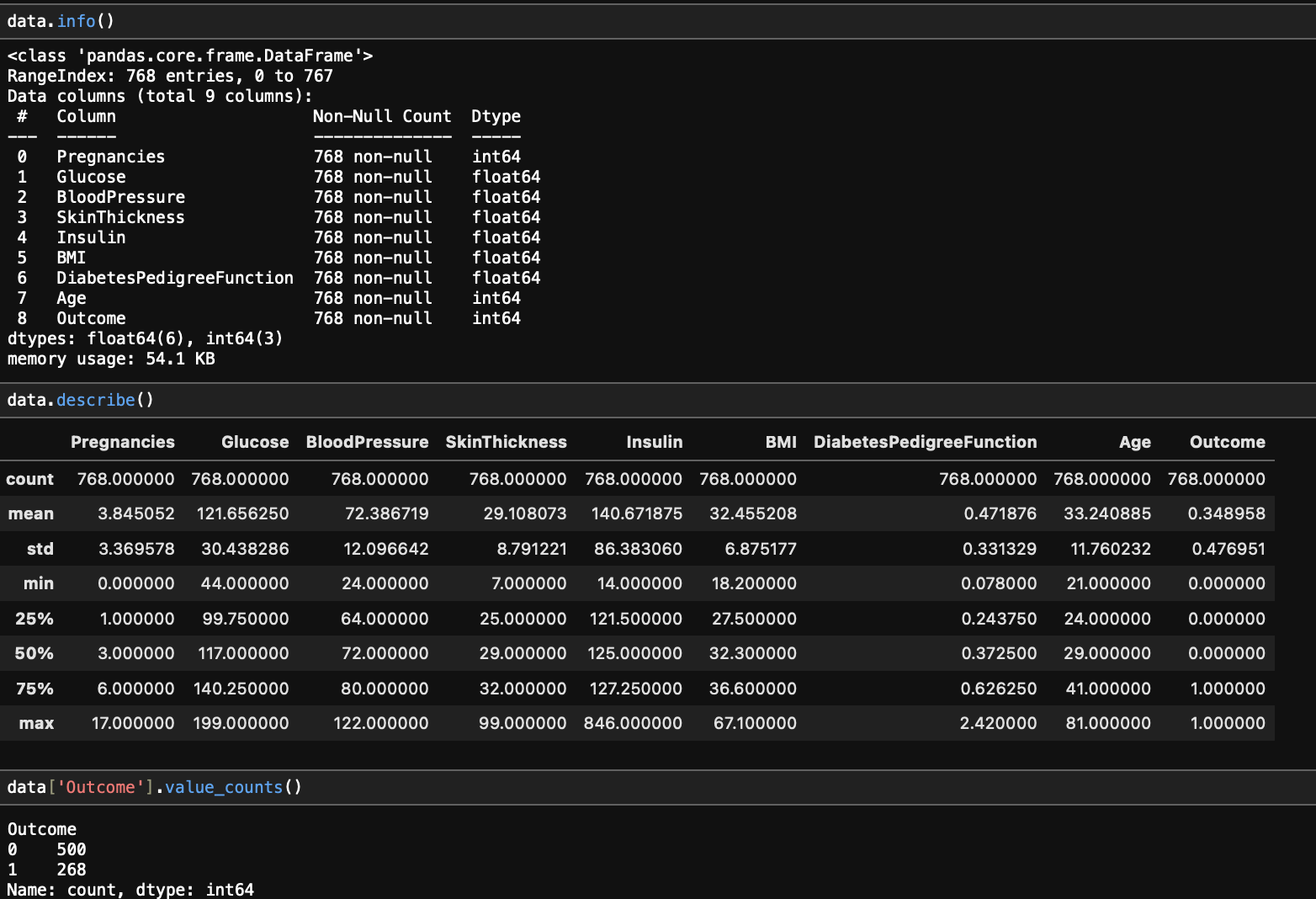


Рис. 1 – Общая информация

df.info()

df.describe()

df['Outcome'].value\_counts()

**4. Гистограммы распределения признаков**

Показываем распределения всех признаков.

Изображение выглядит как График, диаграмма, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 2 – Распределения всех признаков

Изображение выглядит как текст, снимок экрана, График, диаграмма

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3 – Распределения диагнозов

df.hist(bins=20, figsize=(14,10))

plt.suptitle("Распределение признаков")

plt.tight\_layout()

plt.show()

**5. Корреляционная матрица (heatmap)**

Проверяем взаимосвязи между признаками.

Изображение выглядит как текст, снимок экрана, прямоугольный, диаграмма

Содержимое, созданное искусственным интеллектом, может быть неверным.  
Рис. 4 – Корреляционная матрица

plt.figure(figsize=(10, 6))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title("Корреляционная матрица признаков")

plt.show()

**6. Деление на признаки и целевую переменную**

Разделяем на X и y, масштабируем.

X = df.drop('Outcome', axis=1)

y = df['Outcome']

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

**7. Разбиение на обучающую и тестовую выборки**

Делим данные в соотношении 80/20.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X\_scaled, y, test\_size=0.2, random\_state=42)

**8. Настройка гиперпараметров**

Устанавливаем значения гиперпараметров для моделей.

n\_estimators = 100

max\_depth = 5

min\_samples\_split = 2

**9. Обучение моделей**

Обучаем 3 модели: RandomForest, LogisticRegression, GradientBoosting.

rf = RandomForestClassifier(n\_estimators=n\_estimators, max\_depth=max\_depth,

min\_samples\_split=min\_samples\_split, random\_state=42)

rf.fit(X\_train, y\_train)

lr = LogisticRegression(max\_iter=1000, random\_state=42)

lr.fit(X\_train, y\_train)

gb = GradientBoostingClassifier(n\_estimators=n\_estimators, max\_depth=max\_depth,

random\_state=42)

gb.fit(X\_train, y\_train)Изображение выглядит как линия, График, диаграмма, текст

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 5 – Важность признаков по Random Forest

**10. Оценка точности моделей**

Проверяем точность моделей на тестовой выборке.

print("Random Forest:", accuracy\_score(y\_test, rf.predict(X\_test)))

print("Logistic Regression:", accuracy\_score(y\_test, lr.predict(X\_test)))

print("Gradient Boosting:", accuracy\_score(y\_test, gb.predict(X\_test)))

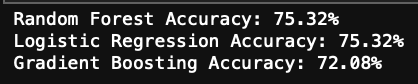


Рис. 6 – Точности моделей

**11. Пример ввода новых данных пациента**

Имитация пользовательского ввода для прогноза.

sample = np.array([[2, 120, 70, 25, 100, 30.5, 0.5, 28]])

sample\_scaled = scaler.transform(sample)

**12. Прогноз вероятности для пациента**

Получаем вероятность диабета от каждой модели.

print("RF вероятность:", rf.predict\_proba(sample\_scaled)[0][1])

print("LR вероятность:", lr.predict\_proba(sample\_scaled)[0][1])

print("GB вероятность:", gb.predict\_proba(sample\_scaled)[0][1])Изображение выглядит как текст, снимок экрана, число, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 7 – Вероятности диабета по моделям

**13. Вывод интерпретации прогноза**

По максимальной вероятности — оценка риска.

probas = [

rf.predict\_proba(sample\_scaled)[0][1],

lr.predict\_proba(sample\_scaled)[0][1],

gb.predict\_proba(sample\_scaled)[0][1],

]

max\_proba = max(probas)

if max\_proba > 0.7:

print("⚠️ Высокий риск!")

elif max\_proba > 0.4:

print("❗ Умеренный риск.")

else:

print("✅ Низкий риск.")



Рис. 8 – Результат оценки риска

**14. Веб-приложение**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 9 – Веб-приложение

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 10 – Сравнение моделей

Заключение

* **Лучшая модель**: **Gradient Boosting**  
  Точность = 75.97%
* **Ансамблевые методы** (Random Forest, Gradient Boosting) **показали наилучшую точность** по сравнению с классической логистической регрессией.
* **Logistic Regression** демонстрирует **стабильность, но слабую зависимость от гиперпараметров**, что ограничивает её возможности в задачах с нелинейными зависимостями.
* **Gradient Boosting** лидирует по метрике точности, что делает его наиболее предпочтительным для данной задачи.