Документация к приложению

«Gucko\_Track»

Версия 1.0

**© Новицкий В.В., 2024**

**Оглавление**

[Введение 3](#_Toc168851431)

[Data Engineering 4](#_Toc168851432)

[Data Mining 6](#_Toc168851433)

[Data Science 8](#_Toc168851434)

[ML Engineering 11](#_Toc168851435)

Введение

Проект «Gluco\_Track» ­ это приложение, предназначенное для оценки значимости параметров, снимаемых мультисенсорной системой, при разработке и выборе модели машинного обучения для определения концентрации глюкозы. Данное приложение охватывает все этапы разработки и анализа в области машинного обучения, обеспечивая комплексный подход к решению задачи.

**Программа состоит из четырёх основных блоков:**

1. *Data Engineering* – обработка и подготовка данных для дальнейшего анализа.

2. *Data Mining* – выявление закономерностей и извлечение значимой информации из данных.

3. *Data Science*– применение научных методов и алгоритмов для анализа данных и подбора подходящей модели машинного обучения.

4. *ML Engineering* – отладка и визуализация лучшей модели машинного обучения.

Эти блоки обеспечивают не только обработку данных, поступающих с мультисенсорной системы, но и проведение корреляционного анализа для определения значимости различных признаков, а также выбор и настройку оптимальной регрессионной модели для решения поставленной задачи. Каждый блок программы способен выполнять независимые операции вне основного сценария программы, что делает приложение гибким и адаптивным к различным новым задачам.

В следующих разделах будет подробно рассмотрена структура и задачи каждого блока, а также продемонстрированы примеры их применения на реальных экспериментальных данных от 2024 года.

После установки для запуска кода необходимо поменять пути к файлам для исправной работы программы.

Data Engineering

Этот этап является основой для последующих процессов анализа данных и включает в себя два основных файла: *«create\_experiment\_dataset.py»* и *«interpolation.py»*. Оба файла предназначены для начальной обработки данных, чтобы подготовить их для дальнейшего использования в модели машинного обучения.

Файл *«create\_experiment\_dataset.py»* объединяет два типа данных: экспериментальные измерения, снятые с оптической пары (АЦП входного и выходного фотодиодов), и временные метки измерений. Также он включает данные из Excel-файла с отчётом инженера по датчикам температуры, что позволяет учитывать как внутренние, так и внешние температурные параметры системы (см. рисунок 1).

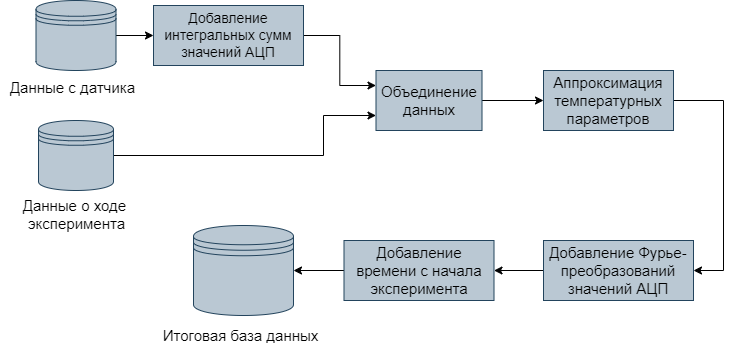


Рисунок 1. Процесс проектирования базы данных

Из-за высокой частоты источника излучения было создано решение для уменьшения количества фиксируемых точек при помощи файла *«interpolation.py»*. Этот файл осуществляет процесс интерполяции, что позволяет уменьшить избыточную дискретизацию и, как следствие, предотвратить перегрузку памяти.

Все обработанные данные сохраняются в папках *«aggregated\_dataframes»* для изначальных файлов и *«aggregated\_dataframe\_interpolated»* для интерполированных результатов экспериментов, которые будут использованы на следующих этапах анализа.

**Особенности данного блока:**

1) Данное приложение рассчитано для оптимального по размерам эксперимента на час. При увеличении времени эксперимента программа обрежет экспериментальный файл до часа. Для расширения времени экспериментов необходимо изменить в файле *«create\_experiment\_dataset.py»* функцию *«add\_num\_sample\_column»* в строчке ***final\_df = final\_df[:300060]***, где размер будет определяться как количество индексов, умноженных на 60, это обусловлено частотой дискретизации прибора, которая равна 5001 Гц.

2) Для составления Excel-отчёта инженером в папке   
«0\_Data» содержится шаблон для заполнения часового эксперимента.

3) Файлы, содержащиеся в папке *«temperature»,* являются результатом экспериментов, где измерения температурных параметров производилось инженером в течение всего времени работы лазера. По результатам данных экспериментов были получены полиномы соответствующие общему характеру изменения температур в течение часового эксперимента. Для расширения времени замера эксперимента рекомендуется провести аналогичный эксперимент и составить новые аппроксимации и полиномы для вычисления температур.

Data Mining

Этот блок предназначен для проведения мультиколинеарного анализа, который выполняется с использованием метода Бэлсли и VIF (фактор инфляции дисперсии). Методика разложения на дисперсионные доли и анализа признаков позволяет определить вклад и влияние каждого признака на систему, а также выявить паразитарные взаимосвязи между признаками.

**Основные файлы и их функции:**

Файл *multicollinearity\_analysis.py* выполняет анализ на мультиколинеарность для всех данных, полученных на этапе Data Engineering. Анализ проводится итеративно: в каждом цикле вычисляются значения VIF для признаков. Признаки с VIF выше 5 и индексом обусловленности (condInd) свыше 10 исключаются. Процесс повторяется до тех пор, пока все признаки не будут удовлетворять указанным критериям.

В конце работы программы создается CSV-файл *removed\_columns\_report.csv*, содержащий список удалённых признаков для каждой итерации анализа. Этот отчёт используется для визуализации и дальнейшего анализа.

Файл *multicollinearity\_visualization.py* отвечает за визуализацию результатов анализа мультиколинеарности. Графики и диаграммы, создаваемые этим скриптом, помогают лучше понять, какие признаки были исключены и как они влияли на модель (см. рисунок 2).

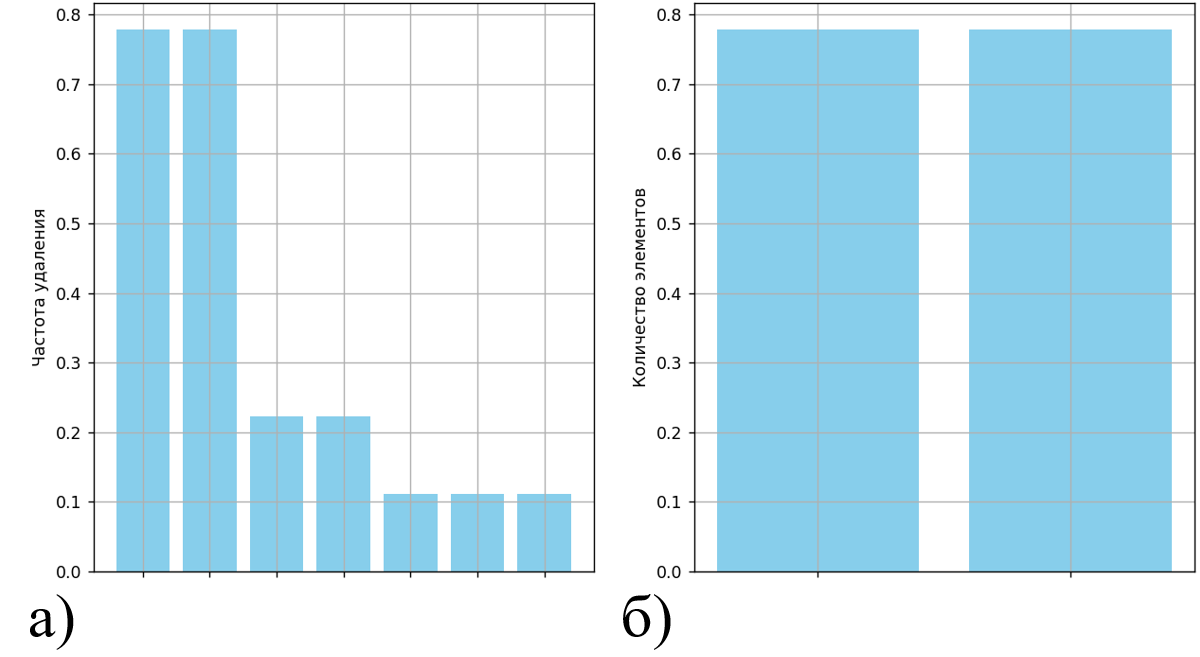
**

Рисунок 2. Визуализация (а) частоты удаления всех признаков и   
(б) признаков, удаляемых в больше, чем 50% случаев.

**Дополнительные шаги:**

После того как проведён первичный анализ мультиколинеарности, файл *shrinkage\_data.py* используется для окончательного удаления признаков. На этом этапе удаляются только те признаки, которые были исключены более чем в 50% случаев анализа. Это позволяет более точно очистить данные, сохранив важные признаки.

Для проверки соответствия данных поставленной задаче используется файл *result\_heatmap.py*. Он строит матрицу корреляции Пирсона (см.рисунок 3), которая позволяет визуально оценить линейные коэффициенты корреляции между признаками. Если результаты показывают, что удалено недостаточно признаков, процесс анализа повторяется до достижения оптимального набора данных.

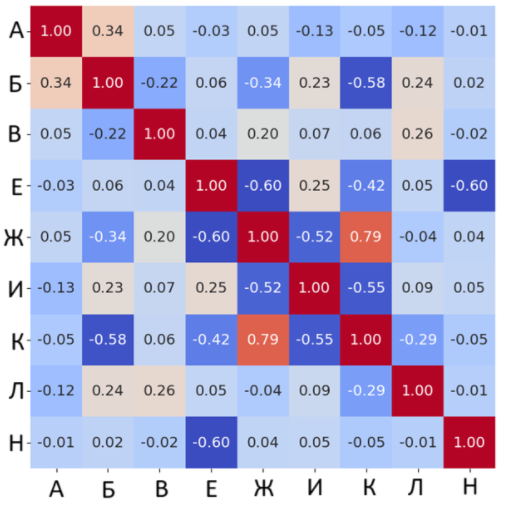


Рисунок 3. Тепловая карта корреляционной матрицы, отображающая степень линейной взаимосвязи между переменными, полученная в результате мультикорреляционного анализа

****Data Science****

Этот блок занимается анализом и выбором наиболее подходящих регрессионных моделей для заданного признакового пространства. Из 40 базовых регрессионных моделей отбираются пять наиболее перспективных, которые затем оптимизируются для достижения наилучших результатов.

**Основные файлы и их функции:**

Файл ***model\_comparison.py*** включает библиотеки и функции для первичной оценки предсказательной способности каждой модели (см. рисунок 4).

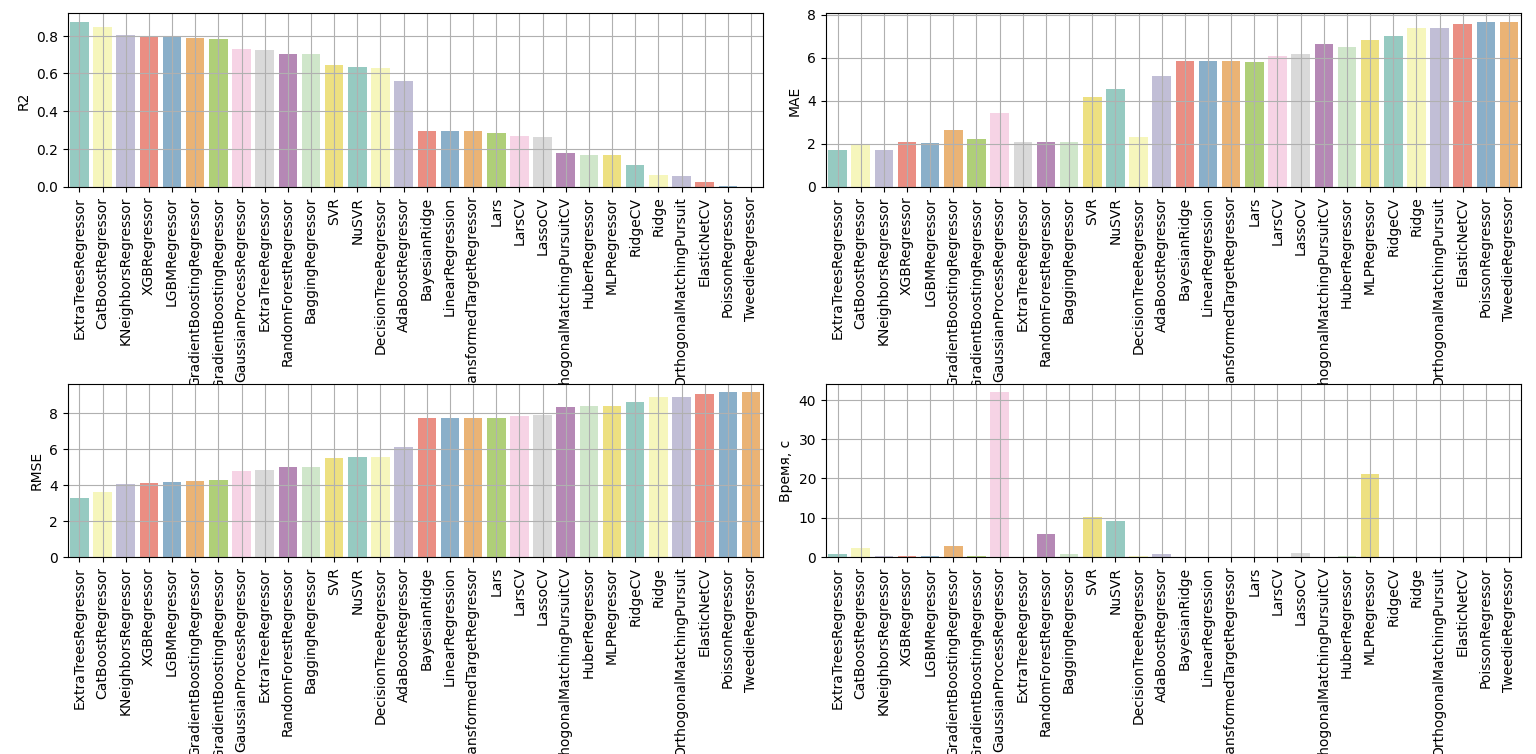


Рисунок 4. Вывод метрик и времени работы 40 моделей машинного обучения

Оценка проводится на теоретических данных, чтобы выбрать наиболее подходящие модели для дальнейшей работы. Данные разделяются следующим образом: 70% данных используется для обучения, 10% – для валидации и 20% – для тестирования модели. Это разделение иллюстрируется на рисунке (см. рисунок 5).

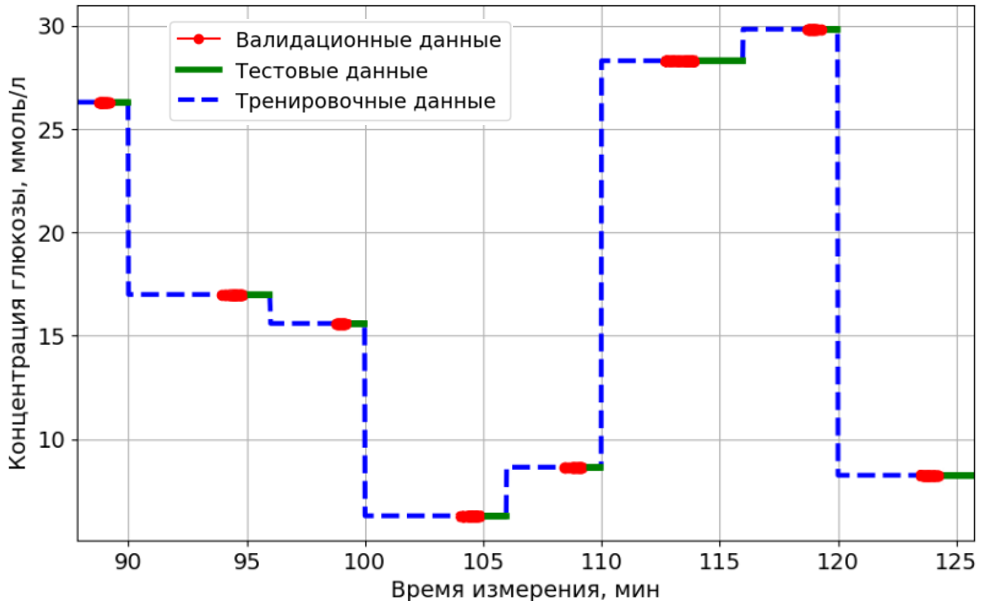


Рисунок 5. Разбиение части одного эксперимента на тренировочные,  
 тестовые и валидационные данные.

Файл **model\_optimization.py** отвечает за оптимизацию пяти лучших моделей, отобранных на предыдущем этапе. Оптимизация включает настройку гиперпараметров моделей для максимального повышения их предсказательной точности. По завершении оптимизации файл *visualization\_models.py* создает визуализацию подробной статистики по четырем основным параметрам, а также сравнительный анализ топ-5 моделей (с. рисунок 6). Это позволяет наглядно оценить преимущества каждой модели.

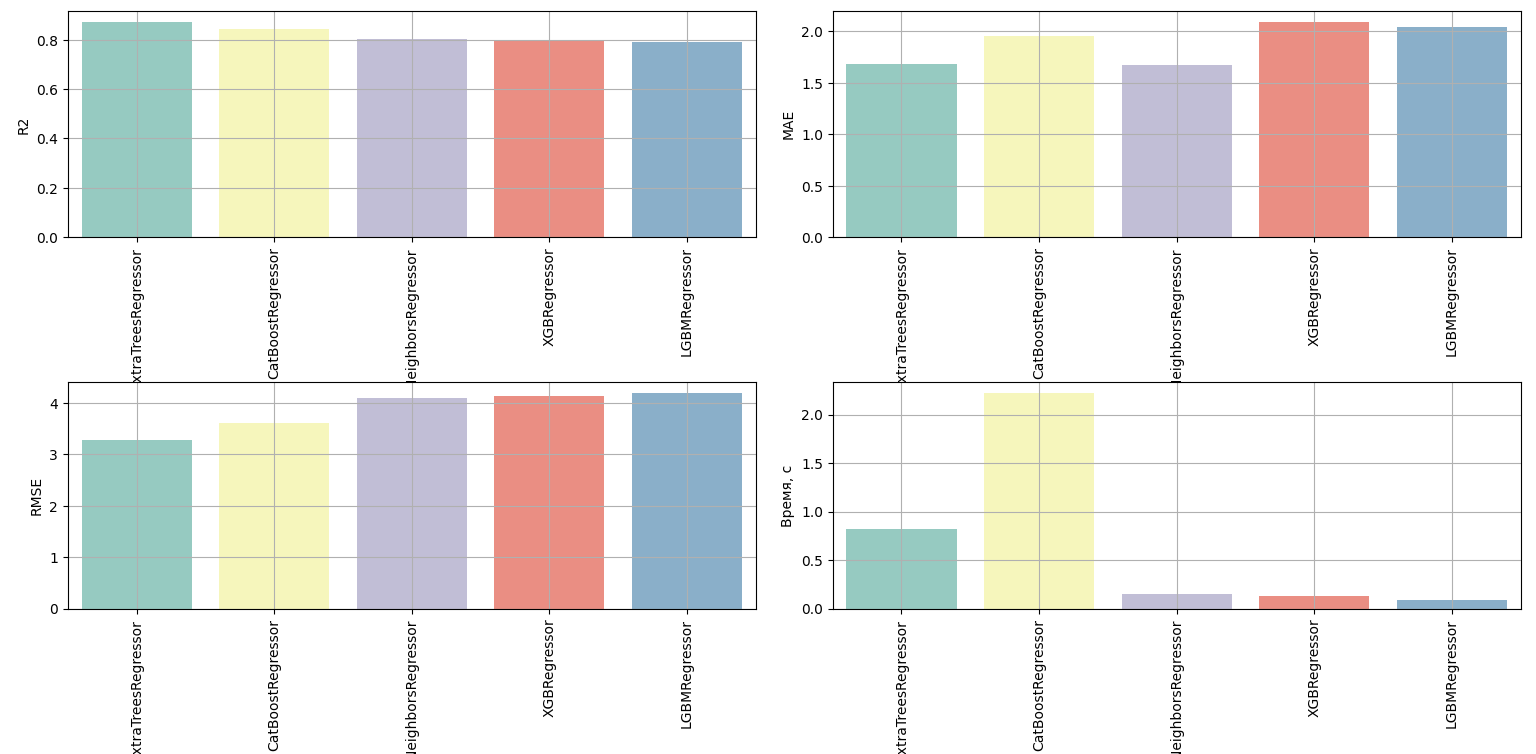
****

Рисунок 6. Вывод метрик и времени работы 5 наилучших моделей машинного обучения

**Дополнительные детали:**

Папка *results\_computation* содержит итоговые разделения на тестовую, тренировочную и валидационную выборки в виде CSV-файлов, а также результаты оптимизаций и модели, подлежащие дальнейшей обработке. В этой директории также находится папка *models\_for\_optimization*, где хранятся модели, разделенные на классы для оптимизации.

Если необходимо добавить новые модели, вышедшие после релиза программы, необходимо обновить два компонента: шаблон для оптимизации Optuna, который находится в папке *models\_for\_optimization* и кортеж *models* с наименованиями всех проверяемых моделей в файле *models\_for\_optimization.py* в функции *evaluate\_models*.

Файл ***visualization\_models.py*** содержит модули для графической визуализации всех 40 моделей машинного обучения (см. рисунок 4), а также детальную визуализацию топ-5 моделей, показавших наилучшие результаты (см. рисунок 6).

По завершении всех этапов создается CSV-файл *optimization\_results.csv*, содержащий результаты оптимизации топ-5 моделей. В этом файле указаны наименования моделей, лучшие значения метрик, по которым проводилась оптимизация, и оптимальные наборы гиперпараметров. Модель, занявшая первое место, используется для дальнейших анализов в блоке ML Engineering.

**ML Engineering**

В этом блоке представлены итоговые и более детальные визуализации результатов, полученных при тестировании модели, показавшей наилучшие результаты.

**Основные файлы и их функции:**

Файл *best\_model\_visualization.py* создает графики, сравнивающие предсказания модели с ожидаемыми значениями, полученными референсным методом во время эксперимента. График предсказаний и измеренных значений: на оси абсцисс указано время измерения в минутах (см. рисунок 7). Это позволяет проводить подробный анализ каждого участка, приближая интересуемые участки графика.

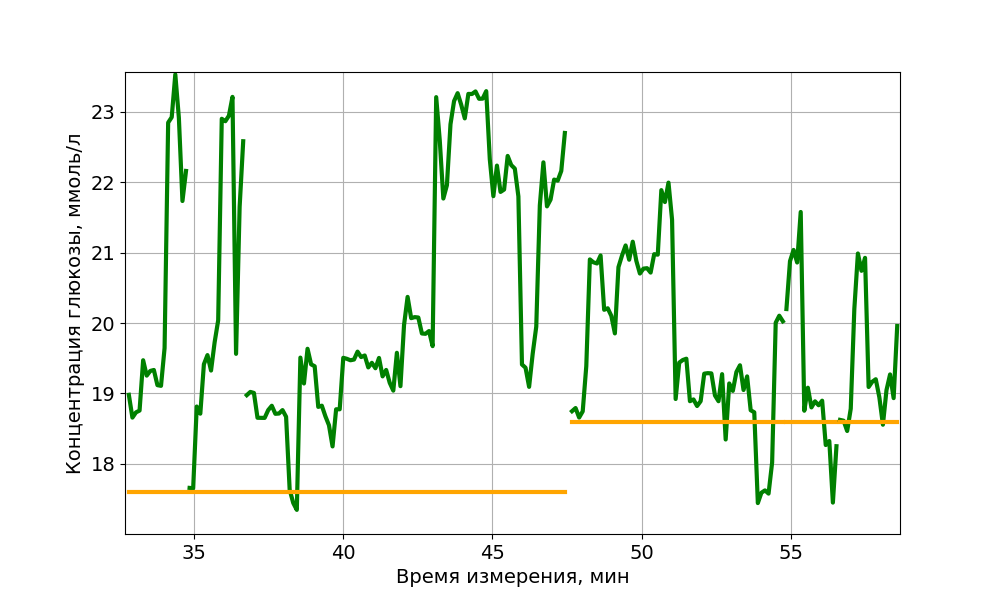


Рисунок 7. График реализации алгоритма определения   
концентрации глюкозы

Вторая визуализация отображает пространство предсказаний относительно ожидаемых значений, что помогает анализировать отклонения концентраций от референсных значений (см. рисунок 8). Линия метода наименьших квадратов: выделенная зеленым цветом, показывает тренд, в то время как линия *y=x* отображает идеальное совпадение предсказаний с теоретическими значениями. Это позволяет легко оценить точность модели в предсказании концентрации глюкозы или другой целевой переменной.

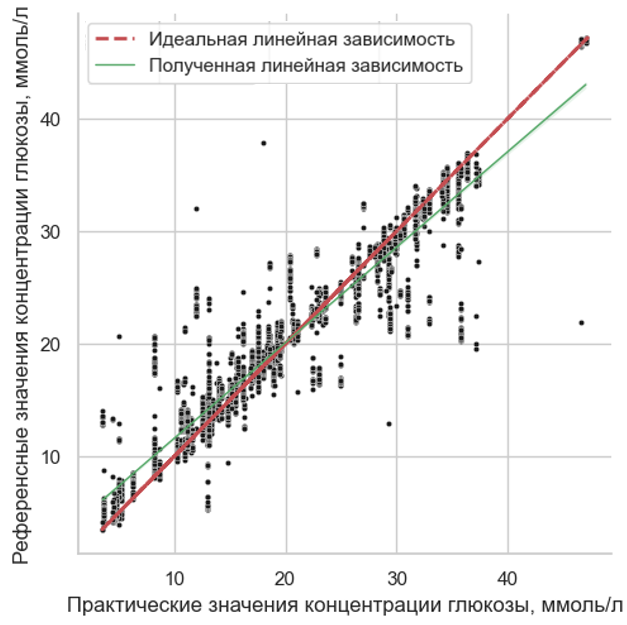
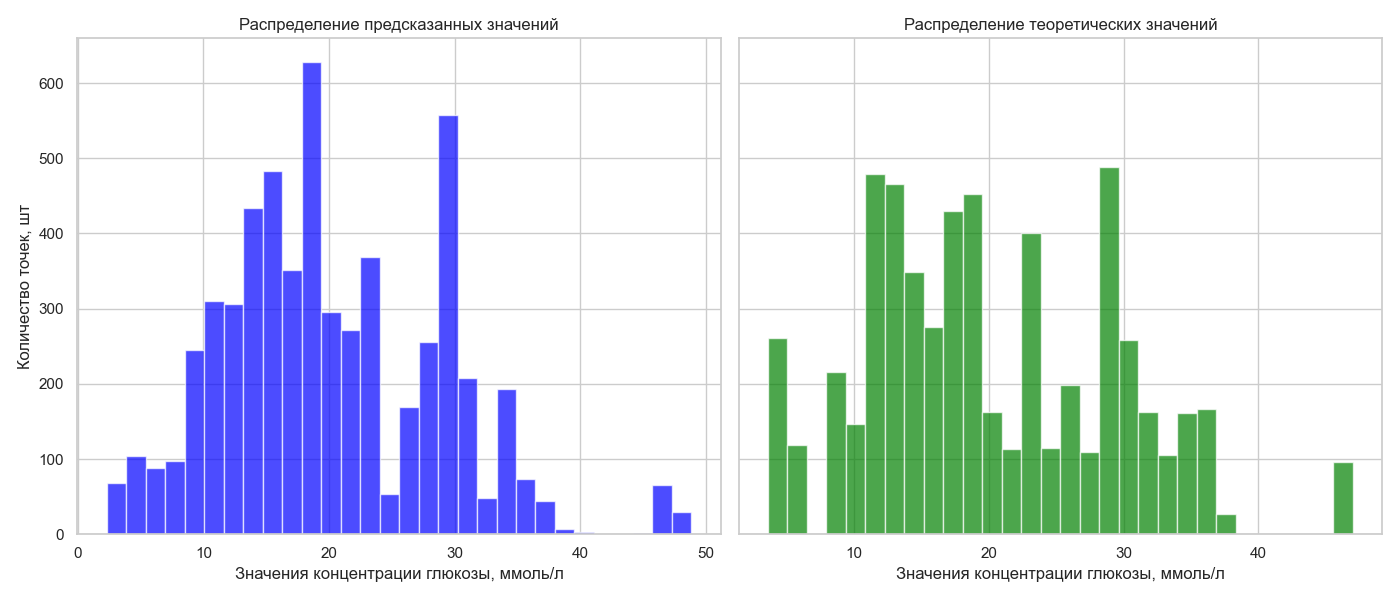


Рисунок 8. Показания прогнозируемых значений от   
ожидаемых концентраций глюкозы

 Распределение концентраций: показывает, как предсказанные значения распределяются относительно исходных данных, что помогает понять точность и форму распределения модели (см. рисунок 9).

Рисунок 9. Распределение предсказанных значений и теоретических значений концентрации глюкозы

 Визуализация зон Кларка или Паркса: показывает, насколько точно модель предсказывает значения в различных зонах, что важно для сертификации алгоритма. Эти зоны демонстрируют правильность и надежность алгоритма, необходимую для медицинской сертификации.

 Анализ разности значений: подробная информация о разнице между тестовыми и предсказанными значениями включает метрики MAE, RMSE, R^2 и MSE. Это позволяет оценить точность модели, определить стандартное отклонение, максимальные и минимальные отклонения (см. рисунок 10).

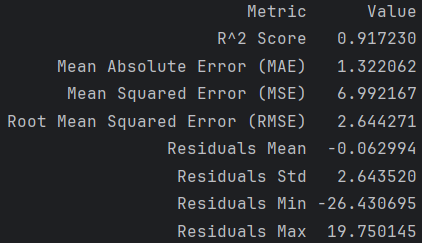


Рисунок 10. Табличный вывод подробной статистике по разности между теоретическими и предсказанными значениями

 Визуализация значимости параметров с помощью модуля SHAP (SHapley Additive exPlanations), что помогает интерпретировать вклад каждого признака в итоговое предсказание модели (см. рисунок 11).



Рисунок 11. Параметры важности признаков в полученной модели  
 гистограммного градиентного бустинга

Данные визуализации помогают понять, как разные параметры влияют на предсказания модели, выявляя общий тренд на различных концентрациях глюкозы. Табличная статистика по метрикам и отклонениям позволяет оценить эффективность и надежность модели.