Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский ядерный университет „МИФИ“»

Курсовая работа
по дисциплине «Классическое машинное обучение»

Выполнил студент 1-го курса
Алексей Котов

2025

1. Постановка курсового проекта

В рамках данного курсового проекта поставлена задача создания моделей, способных прогнозировать эффективность химических соединений против вируса гриппа и определять наиболее перспективные кандидаты для дальнейших лабораторных испытаний и разработки медикаментов. Исходные данные включают информацию о 1000 химических соединениях, для каждого из которых известны показатели IC50, CC50 и SI.

Основные этапы работы:

1. Провести исследовательский анализ данных, определить возможные выбросы, по необходимости заполнить пропуски, выявить аномалии и закономерности, которые могут повлиять на построение эффективных моделей.

2. Создать и сравнить несколько регрессионных моделей для предсказания непрерывных целевых переменных:

- Регрессия для IC50;
- Регрессия для CC50;
- Регрессия для SI.

Для каждой задачи подобрать различные алгоритмы, выполнить настройку гиперпараметров и оценить качество моделей по метрикам RMSE, MAE и R2.

3. Построить и оценить классификационные модели на основе бинарных меток, сформированных по следующим критериям:

- превышение медианного значения IC50;
- превышение медианного значения CC50;
- превышение медианного значения SI;
- превышение значения SI порога 8.

В каждом случае протестировать разные алгоритмы, оптимизировать гиперпараметры и оценить модели с помощью accuracy, f1-score, precision и recall

4. Провести сравнительный анализ всех построенных моделей по выбранным метрикам и обосновать выбор наиболее эффективных для каждой из задач.

2. Исследовательский анализ данных,

2.1 Обработка пропущенных значений

При первичном осмотре набора данных было выявлено наличие пропусков в следующих столбцах:

- MaxPartialCharge — 3 пропуска
- MinPartialCharge — 3 пропуска
- MaxAbsPartialCharge — 3 пропуска
- MinAbsPartialCharge — 3 пропуска
- BCUT2D_MWHI — 3 пропуска
- BCUT2D_MWLOW — 3 пропуска
- BCUT2D_CHGHI — 3 пропуска
- BCUT2D_CHGLO — 3 пропуска
- BCUT2D_LOGPHI — 3 пропуска
- BCUT2D_LOGPLOW — 3 пропуска
- BCUT2D_MRHI — 3 пропуска
- BCUT2D_MRLOW — 3 пропуска

Текстовых колонок в наборе не обнаружено. Для устранения пропусков была использована методика заполнения на основе алгоритма KNNImputer.

2.2 Устранение константных признаков

При подготовке данных мы проверили каждый столбец на наличие единственного уникального значения. Такие «константные» признаки не вносят никакой информации и лишь увеличивают размерность данных. Были обнаружены следующие 17 константных столбцов:

NumRadicalElectrons, SMR_VSA8, SlogP_VSA9, fr_N_O, fr_SH, fr_azide, fr_barbitur, fr_benzodiazepine, fr_diazo, fr_dihydropyridine, fr_isocyan, fr_isothiocyan, fr_lactam, fr_nitroso, fr_phos_acid, fr_phos_ester, fr_prisulfonamd, fr_thiocyan.

Удаление этих признаков позволило сократить избыточность данных и ускорить последующую работу моделей.

2.3 Дубликаты и валидация целевых переменных

После очистки от константных столбцов мы проверили наличие повторяющихся строк. Полные дубликаты могут искажать результаты обучения и приводить к переобучению. С помощью метода data.duplicated().sum() мы убедились, что в наборе нет ни одной повторяющейся записи.

Затем важно было удостовериться, что в целевых переменных IC50, CC50 и SI отсутствуют нулевые или отрицательные значения, так как они не имеют биологического смысла и могут сломать алгоритмы обучения. Для каждой из трёх метрик мы подсчитали число неположительных наблюдений: результат показал, что все значения строго положительные, что позволило нам продолжить анализ без удаления дополнительных записей.

2.4 Удаление высококоррелированных признаков

При большом числе признаков часто возникает мультиколлинеарность, когда некоторые переменные оказываются почти линейно зависимы друг от друга. Это может привести к нестабильности коэффициентов в регрессионных моделях и усложнить интерпретацию. Чтобы избежать этого, мы воспользовались классом DropCorrelatedFeatures с порогом корреляции 0.9. После автоматического отбора и исключения избыточных признаков размерность датасета уменьшилась, при этом была сохранена максимальная часть информации.

2.5 Анализ распределений целевых переменных

2.5.1 Первичный анализ: гистограммы и Q–Q-графики

Сначала мы построили гистограммы и Q–Q-графики для исходных значений IC50, CC50 и SI, чтобы визуально оценить форму распределений и наличие отклонений от нормального закона.

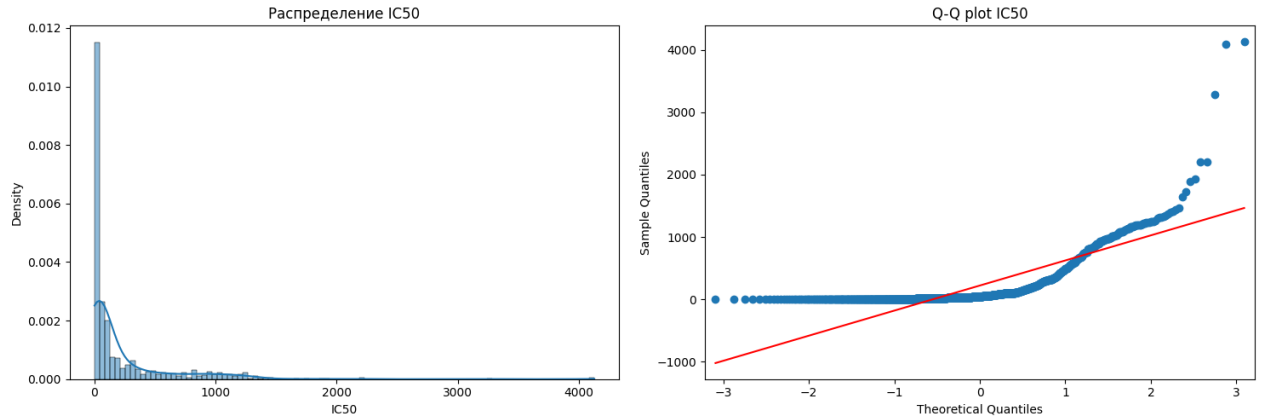

Рисунок 2.5.1.1. Распределение значений IC50 и его Q–Q-график

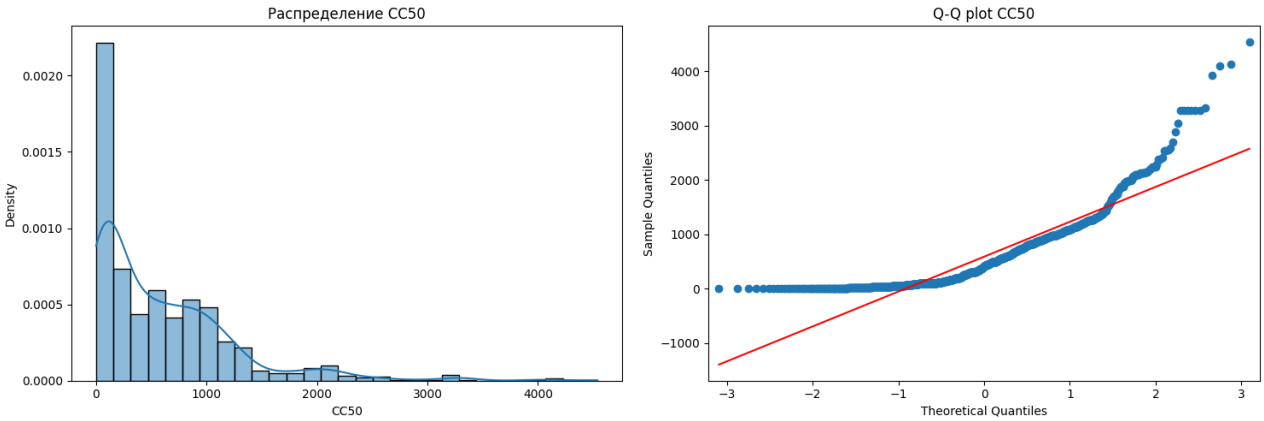

Рисунок 2.5.1.2. Распределение значений CC50 и его Q–Q-график

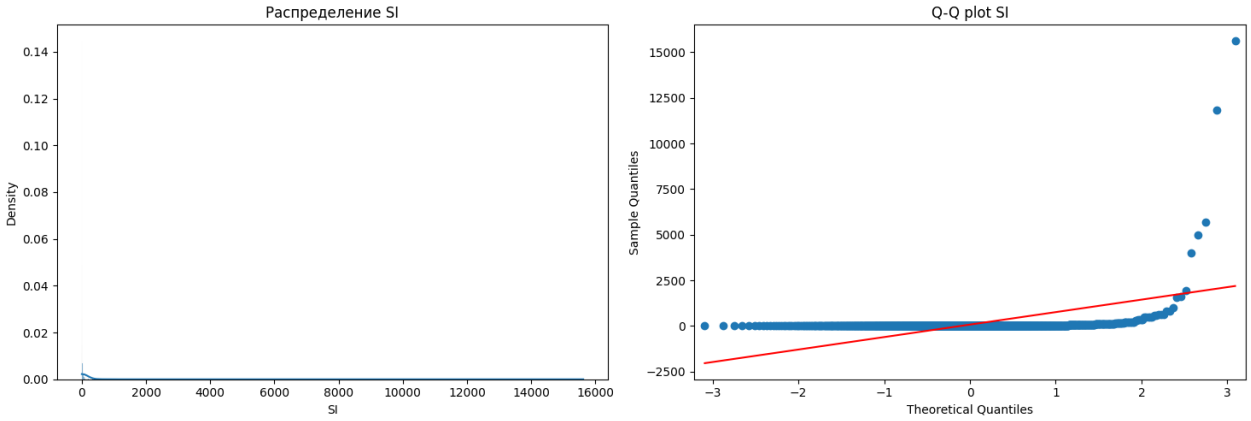

Рисунок 2.5.1.3. Распределение значений SI и его Q–Q-график

По результатам теста Шапиро–Уилка ($p < 0.001$ для всех трёх метрик), а также по сильной правосторонней асимметрии и выраженному пику на гистограммах, мы пришли к заключению о значительном отклонении распределений от нормальных.

2.5.2 Лог-преобразование

Чтобы уменьшить выраженность правосторонних хвостов и выровнять разброс значений целевых переменных, мы применили преобразование вида log1p(x). Этот метод имеет два ключевых преимущества:

1. **Сглаживание сильного скошенного распределения.** Данные по IC50, CC50 и SI изначально показывали ярко выраженный правый хвост: небольшая часть наблюдений имела крайне высокие значения, которые искажали средние и дисперсию. Логарифмическое преобразование сдвигает большие значения относительно меньших, «сжимая» хвосты и приближая форму распределения к более симметричной.

2. **Корректная работа с нулевыми и малыми значениями.** Поскольку в биологических измерениях иногда встречаются нули или очень малые положительные значения, обычный натуральный логарифм мог бы привести к неопределённостям. Добавление единицы внутри аргумента — метод log1p(x) — гарантирует, что все точки данных остаются допустимыми для преобразования и минимизирует искажения в области малых значений.

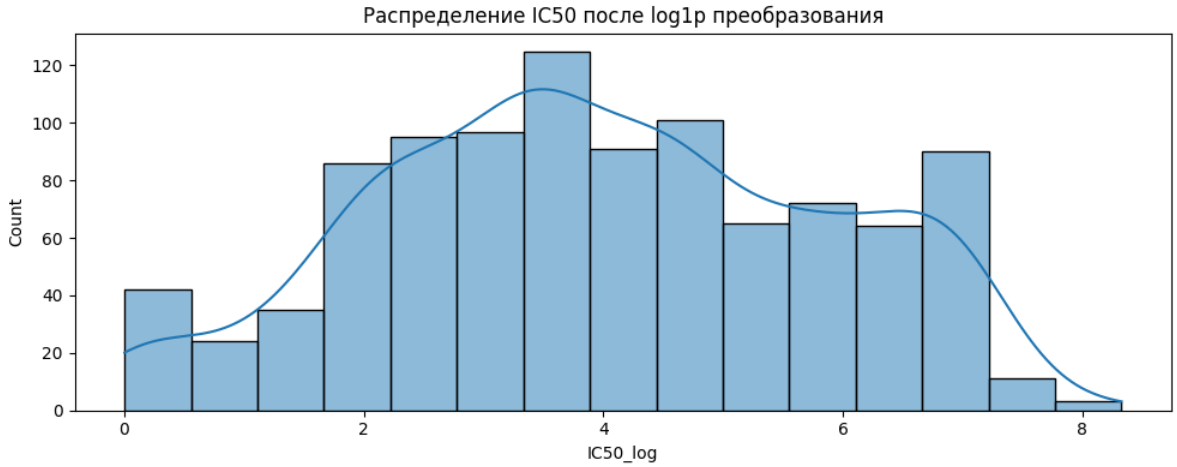

Рисунок 2.5.2.1. График распределение IC50 после log1p преобразования

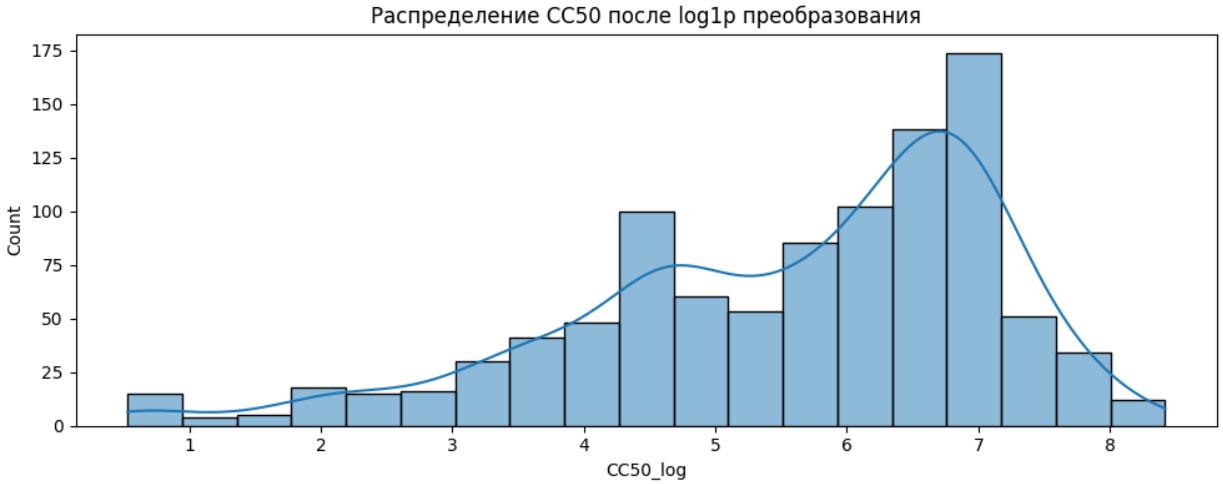

Рисунок 2.5.2.2. График распределение CC50 после log1p преобразования

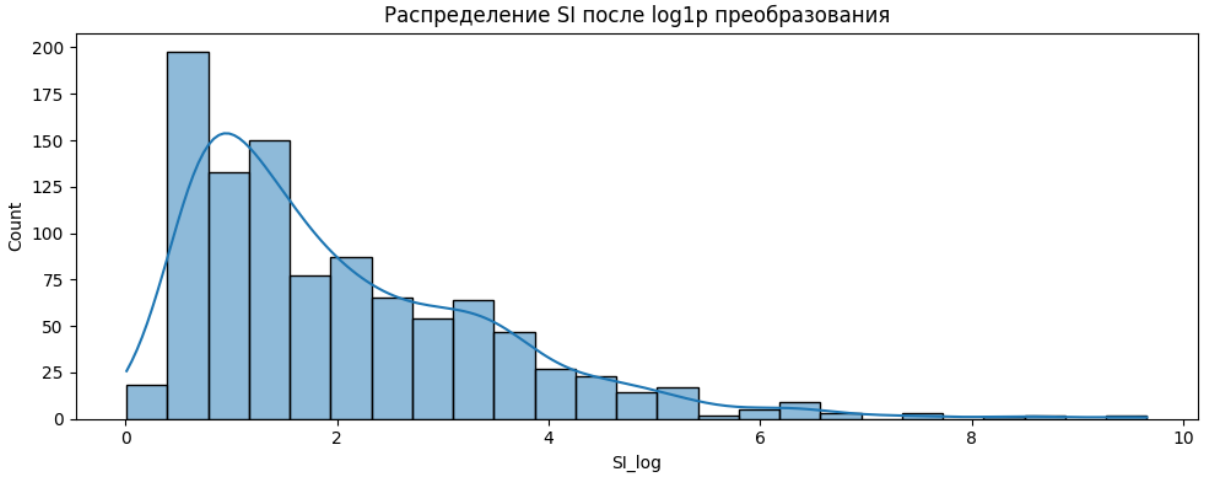

Рисунок 2.5.2.3. График распределение SI после log1p преобразования

Заметно, что у IC50_log1p и CC50_log1p асимметрия снизилась, распределения стали более компактными. В то же время SI_log1p всё ещё содержит многочисленные выбросы вправо.

2.6 Анализ выбросов и хвостов после лог-преобразования

2.6.1 Boxplot-графики исходных лог-данных

После лог-преобразования мы построили боксплоты для трёх целевых переменных и рассчитали для каждой длины «левых» и «правых» хвостов, опираясь на 5-й и 95-й процентили:

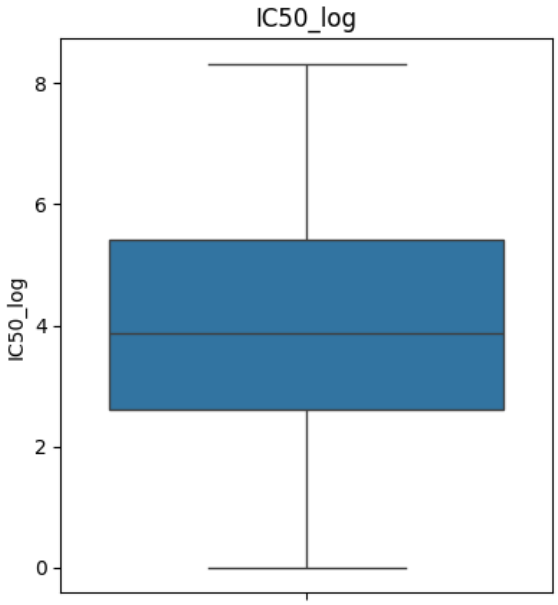

Рисунок 2.6.1.1. Boxplot-график IC50_log1p

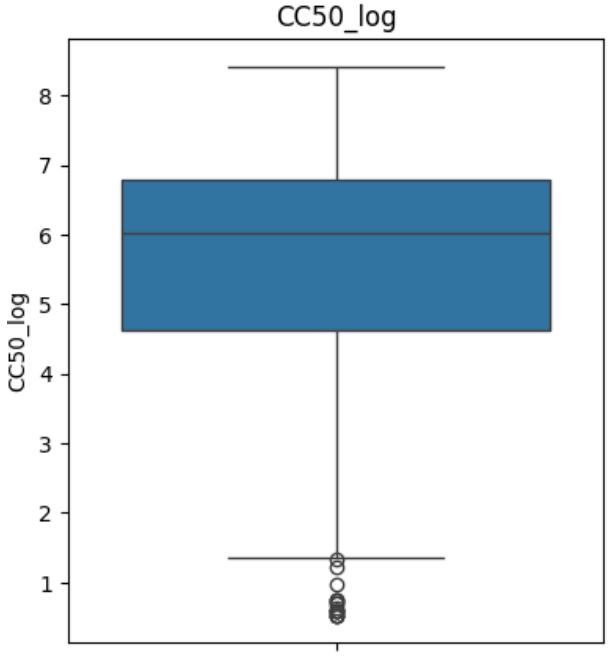

Рисунок 2.6.1.2. Boxplot-график CC50_log1p

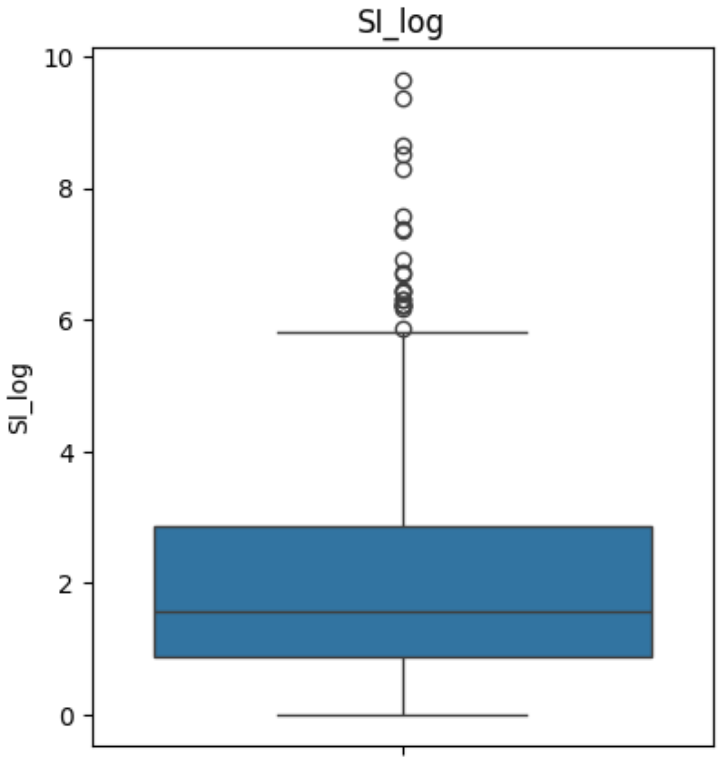

Рисунок 2.6.1.3. Boxplot-график SI_log1p

Переменная	**5% квантиль**	**Медиана**	**95% квантиль**	**Левый хвост (медиана–q5)**	**Правый хвост (q95–медиана)**	**Отношение (правый/левый)**
IC50_log1p	0.8753	3.8625	6.9826	2.9872	3.1201	1.0445
CC50_log1p	2.3785	6.0211	7.5691	3.6426	1.5480	0.4250
SI_log1p	0.6869	1.5782	4.7418	0.8913	3.1636	3.5494

Из данных показателей видно:

- **IC50_log1p**: отношение хвостов близко к 1 (≈ 1.04), то есть распределение практически симметричное. Явных выбросов для удаления не требуется, поэтому мы оставляем весь диапазон значений без дальнейшей очистки.
- **CC50_log1p**: правый хвост (1.55) значительно короче левого (3.64), отношение ≈ 0.43 указывает на смещение распределения влево. Чтобы выровнять хвосты и избавиться от чрезмерно мелких значений, мы отклоняем все наблюдения

ниже 5-го перцентиля (2.3785). Это удаление позволит модели не «зацикливаться» на аномально низких значениях и улучшит качество прогноза.

- **SI_log1p**: правый хвост (3.16) более чем в три раза длиннее левого (0.89), отношение ≈ 3.55 подтверждает наличие экстремальных высоких значений. Для очистки данных мы применяем классическое правило «1.5 × IQR»: вычисляем IQR = Q3–Q1 и удаляем все точки, превышающие Q3 + 1.5 × IQR. Это эффективно убирает самые значительные выбросы, сохраняя при этом общую форму распределения.

Таким образом, после построения boxplot-графиков и расчёта хвостовых метрик мы приняли следующие решения:

1. **IC50_log1p** — оставить без изменений.
2. **CC50_log1p** — отсечь значения < 5-го перцентиля.
3. **SI_log1p** — применить фильтрацию по Q3 + 1.5 × IQR.

Эти шаги обеспечивают сбалансированность распределений и готовят данные к финальному этапу моделирования.

2.6.2 Повторный анализ после удаления выбросов

После отсечения крайних значений в CC50_log1p и применения правила «1.5 × IQR» к SI_log1p мы вновь построили boxplot-графики и гистограммы, чтобы убедиться в эффективности проведённой очистки:

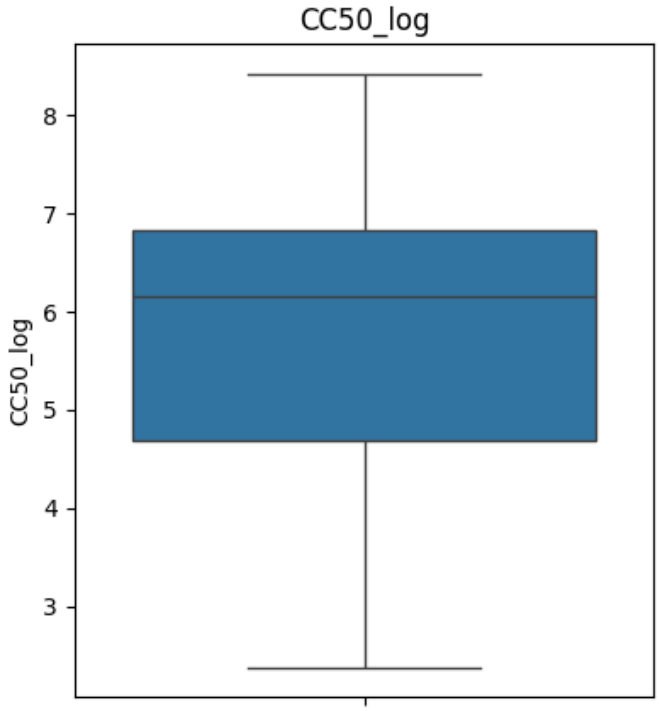

Рисунок 2.6.2.1. Boxplot-график CC50_log1p после удаления значений ниже 5%-го перцентиля

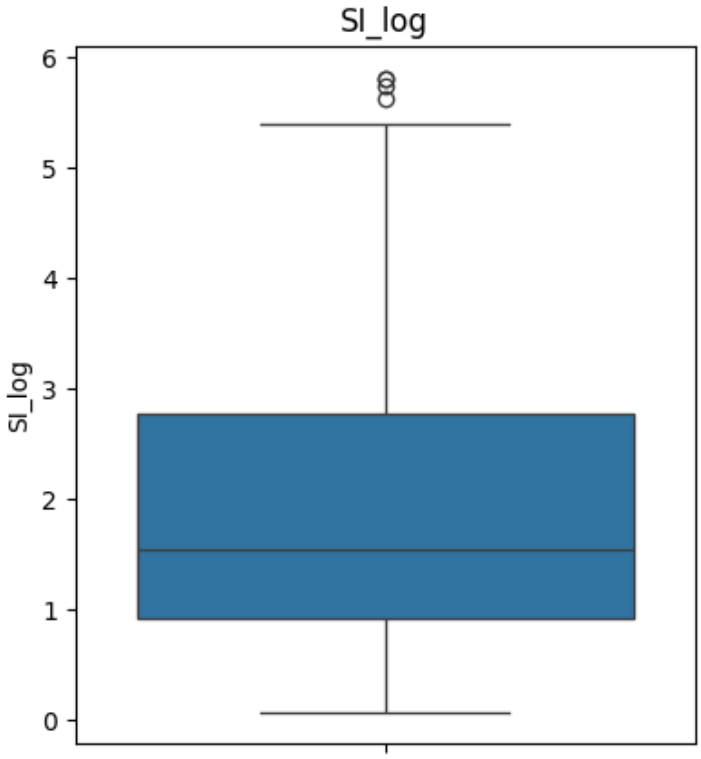

Рисунок 2.6.2.2. Boxplot-график SI_log1p после отсева по правилу 1.5 × IQR

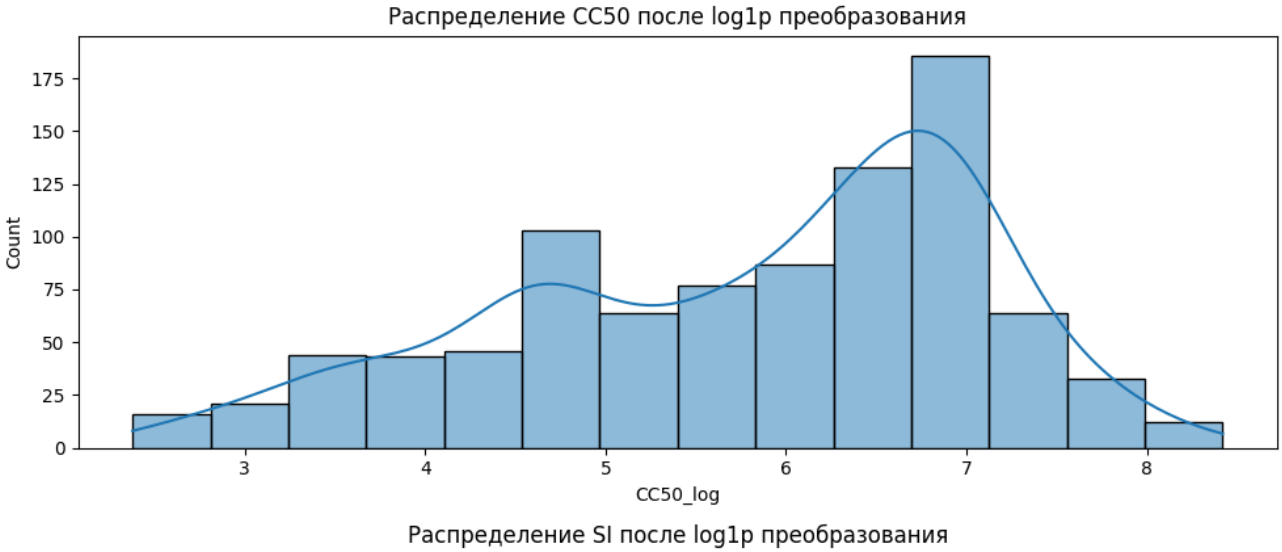

Рисунок 2.6.2.3. Гистограмма распределения CC50_log1p после очистки

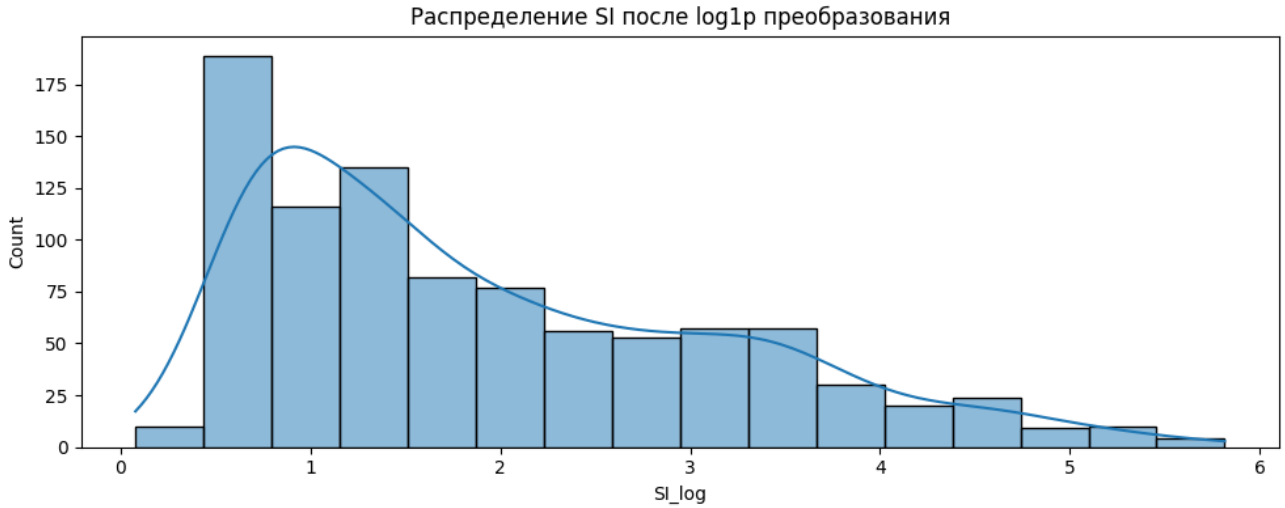

Рисунок 2.6.2.4. Гистограмма распределения SI_log1p после очистки

Визуальный и количественный анализ показывает:

- В **CC50_log1p** после исключения самых мелких значений (<5%-го перцентиля) гистограмма приобрела более «плотную» форму без «длинного» левого хвоста, а boxplot-график не выявляет аномалий.
- В **SI_log1p** число экстремальных точек существенно сократилось, а хвосты стали приемлемой длины: гистограмма демонстрирует адекватный разброс, а boxplot-график показывет, что отсутствует чрезмерная концентрация выбросов.

Таким образом, после очищения данных все три целевые переменные демонстрируют более симметричные и компактные распределения, что создаёт

надёжную основу для дальнейшего построения регрессионных и классификационных моделей.

3. Создание моделей регрессии

На основании проделанной предобработки целевые метрики после удаления экстремальных хвостов по-прежнему остаются далёкими от нормального распределения. Это накладывает ограничение на применение линейных моделей с предпосылкой нормальности остатков и вынуждает сосредоточиться на методах, устойчивых к отклонениям от нормального закона и выбросам.

В нашем подходе мы остановились на алгоритмах на основе деревьев решений и бустинга, которые не требуют предварительной масштабной трансформации данных и сами по себе эффективно справляются с коррелированными и шумными признаками. Были выбраны следующие модели:

- XGBoost;
- Random Forest;
- CatBoost;
- Gradient Boosting;
- HistGradient Boosting;
- Extra Trees.

При этом мы сознательно отказались от методов понижения размерности (например PCA), поскольку эти алгоритмы успешно обходятся с избыточностью и высоким числом признаков без потери качества.

Пайплайн эксперимента на каждой из трёх целевых переменных включает три стадии:

1. **Базовое обучение** – все модели обучаются на тренировочном подмножестве без какой-либо настройки гиперпараметров моделей, после чего по тестовой выборке рассчитывается базовое значение метрик.
2. **Грубый поиск по сетке (RandomizedGridSearch)** – для каждого алгоритма выполняется случайный перебор комбинаций гиперпараметров, результаты сводятся в отдельную таблицу с метриками.

3. **Тонкая настройка Optuna** – модель, показавшая наилучшие результаты в двух предыдущих шагах, подвергается оптимизации с помощью библиотеки Optuna. В качестве целевой метрики при подборе гиперпараметров мы используем среднеквадратическую ошибку RMSE, поскольку:

- Измеряется в тех же единицах, что и цель, что упрощает интерпретацию;
- Не зависит от среднего уровня целевой переменной и одинаково применим к разным задачам;
- Максимально соответствует задаче минимизации реальных отклонений прогнозов.

В результате на выходе optuna_tuning возвращает окончательную модель, обученную на полном тренировочном наборе с оптимальными параметрами, а также статистку, которая показывает, насколько лучше стала модель после этого шага.

Такой многоступенчатый подход гарантирует честное и всестороннее сравнение алгоритмов, тщательный подбор гиперпараметров на валидации и объективную оценку качества на отложенной выборке.

3.1 Создание модели регрессии для IC50

Оценка моделей с параметрами по умолчанию

На первом этапе были обучены модели с параметрами по умолчанию. Лучшую производительность показала модель Extra Trees, достигнув наименьшего значения RMSE (1.1079) и самого высокого R2 (0.5563), что свидетельствует о высоком качестве предсказаний. В то же время XGBoost продемонстрировал наихудший результат по всем метрикам, что объясняется отсутствием настройки и возможной переоценкой сложности модели на этом этапе.

Model	RMSE	MAE	R2	Hyperparameters
XGBoost	1.2472	0.9072	0.4378	False
HistGradient Boosting	1.1718	0.8873	0.5037	False
Random Forest	1.1515	0.8669	0.5208	False
Gradient Boosting	1.1478	0.8887	0.5238	False
CatBoost	1.1412	0.8660	0.5293	False
Extra Trees	1.1079	0.8151	0.5563	False

Результаты после RandomizedSearch

После настройки гиперпараметров все модели показали улучшение, особенно заметное у XGBoost, который вышел на первое место с RMSE = 1.0957 и наибольшим R2 = 0.5661. Результаты демонстрируют высокую зависимость качества предсказаний от гиперпараметров и необходимость их подбора. Extra Trees хоть и осталась в тройке лидеров, не показала существенного улучшения по сравнению с дефолтной версией.

Model	RMSE	MAE	R2	Hyperparameters
XGBoost	1.0957	0.8650	0.5661	True
CatBoost	1.1060	0.8742	0.5579	True
Extra Trees	1.1132	0.8895	0.5521	True
Gradient Boosting	1.1216	0.8758	0.5453	True
Random Forest	1.1384	0.8880	0.5316	True
HistGradient Boosting	1.1386	0.9024	0.5315	True

Общая сводка

Агрегированная таблица демонстрирует общий прогресс моделей. XGBoost преобразовался из худшего варианта в лидера, что подчёркивает важность подбора гиперпараметров. При этом Extra Trees хоть и остаётся сильным претендентом по MAE, не улучшила RMSE и R2, что может свидетельствовать об ограниченном потенциале модели в рамках текущего подхода.

Model	RMSE	MAE	R2	Hyperparameters
XGBoost	1.0957	0.8650	0.5661	True
CatBoost	1.1060	0.8742	0.5579	True
Extra Trees	1.1079	0.8151	0.5563	False
Extra Trees	1.1132	0.8895	0.5521	True
Gradient Boosting	1.1216	0.8758	0.5453	True
Random Forest	1.1384	0.8880	0.5316	True
HistGradient Boosting	1.1386	0.9024	0.5315	True
CatBoost	1.1412	0.8660	0.5293	False
Gradient Boosting	1.1478	0.8887	0.5238	False
Random Forest	1.1515	0.8669	0.5208	False
HistGradient Boosting	1.1718	0.8873	0.5037	False
XGBoost	1.2472	0.9072	0.4378	False

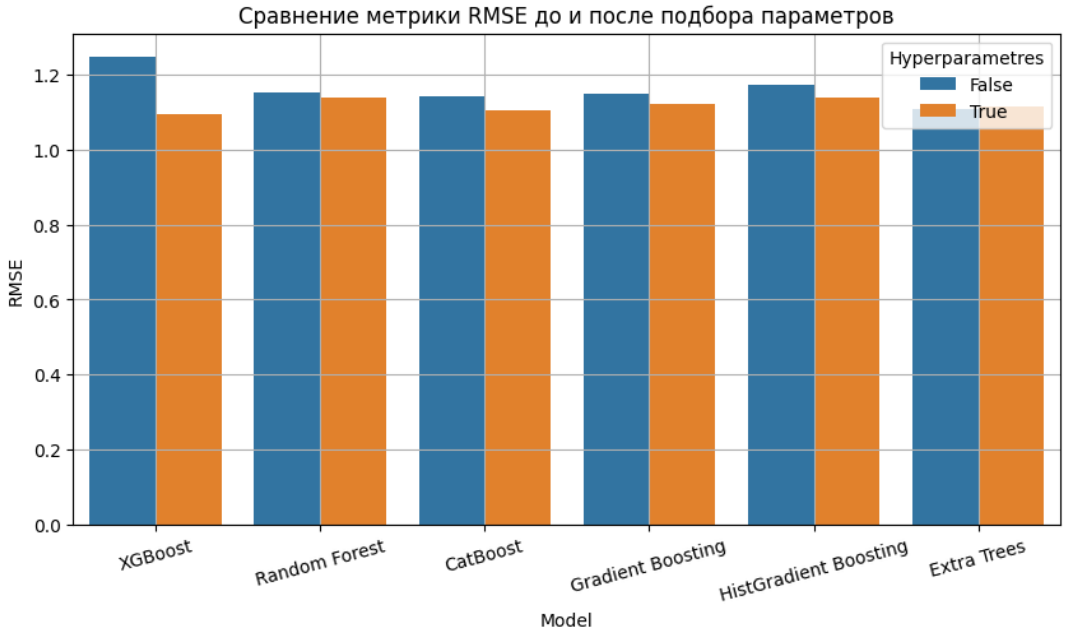

Рисунок 3.1.1. Сравнение метрики RMSE до и после подбора гиперпараметров

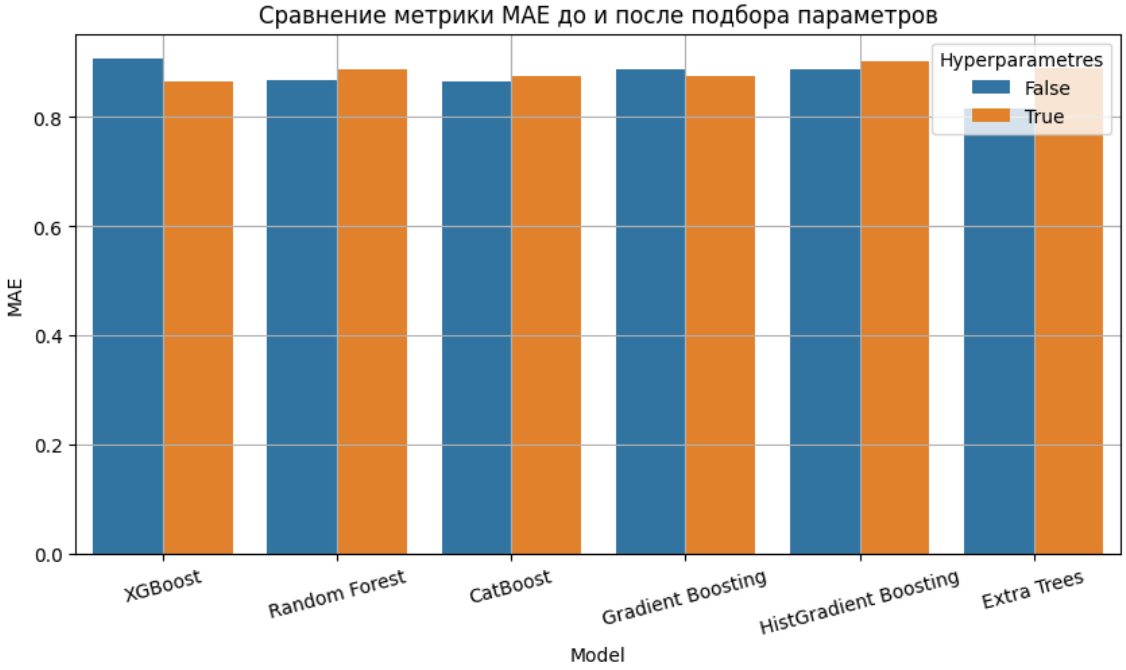

Рисунок 3.1.2. Сравнение метрики MAE до и после подбора гиперпараметров

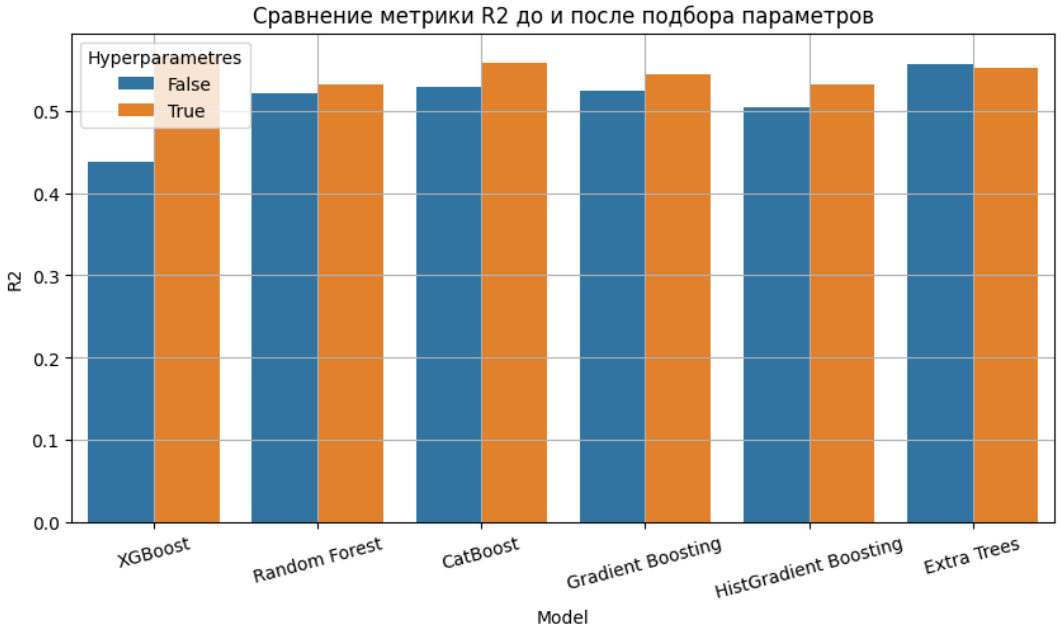

Рисунок 3.1.3. Сравнение метрики R2 до и после подбора гиперпараметров

Сводка по метрикам

Метрика: RMSE

- Лучшая модель: XGBoost → RMSE = 1.0957
- Худшая модель: XGBoost → RMSE = 1.2472
- Без улучшений: Extra Trees

Метрика: MAE

- Лучшая модель: Extra Trees → MAE = 0.8151
- Худшая модель: XGBoost → MAE = 0.9072
- Без улучшений: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Метрика: R2

- Лучшая модель: XGBoost → R2 = 0.5661
- Худшая модель: XGBoost → R2 = 0.4378
- Без улучшений: Extra Trees

Финальная настройка с Optuna

Оптимизация с помощью Optuna позволила незначительно, но стабильно улучшить все ключевые метрики. Хотя прирост невелик, он подтверждает, что выбранные параметры находятся вблизи оптимума. Модель XGBoost окончательно закрепила своё лидерство.

Финальные метрики после Optuna:

RMSE	MAE	R2
1.0935	0.8467	0.5678

Визуализация чётко отражает поэтапное улучшение модели. Каждая стадия от дефолтной до финальной — вносила вклад в повышение качества. Особенно заметен скачок от базовой модели к RandomizedSearch. Optuna принес точечные донастройки, укрепив результат.

Рисунок 3.1.4. Изменение метрики RMSE по этапам

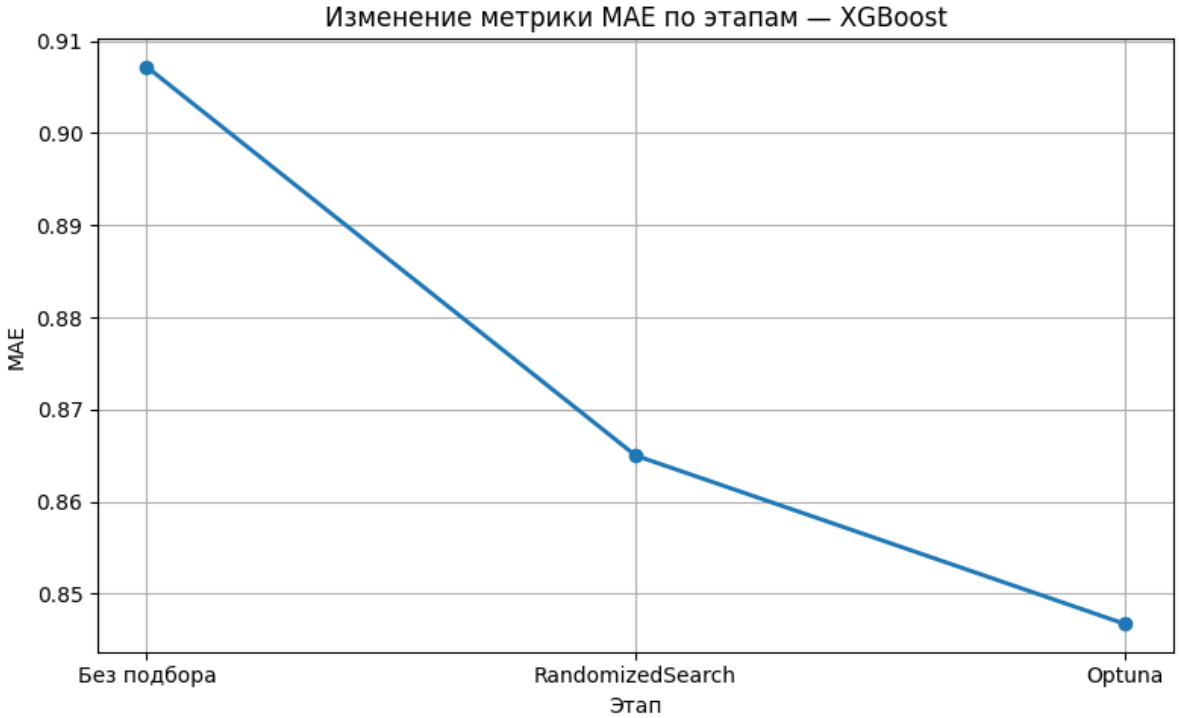

Рисунок 3.1.5. Изменение метрики MAE по этапам

Рисунок 3.1.6. Изменение метрики R2 по этапам

Вывод по модели IC50

На основании полученных результатов, оптимальным выбором модели для задачи предсказания IC50 является XGBoost с подбором гиперпараметров, выполненным с помощью Optuna. Модель показала наилучшие значения по метрикам RMSE и R2, а её финальные настройки обеспечили стабильное улучшение качества предсказаний.

3.2 Создание модели регрессии для CC50

На начальном этапе обучения модели тестировались без подбора гиперпараметров. Лучшую производительность показала модель Extra Trees, продемонстрировав минимальное значение RMSE (0.7939) и наивысший R2 (0.6074), что указывает на высокое качество предсказаний. Наиболее слабый результат выдал XGBoost, что, вероятно, связано с чрезмерной сложностью модели без соответствующей настройки.

Model	RMSE	MAE	R2	Hyperparametres
XGBoost	0.8858	0.6031	0.5113	False
HistGradient Boosting	0.8661	0.6000	0.5328	False
Gradient Boosting	0.8525	0.6182	0.5473	False
CatBoost	0.8261	0.5923	0.5749	False
Random Forest	0.8221	0.5685	0.5790	False
Extra Trees	0.7939	0.5362	0.6074	False

Результаты после RandomizedSearch

После подбора гиперпараметров через RandomizedSearch наблюдается улучшение в производительности моделей. Особенно хорошо себя проявил XGBoost, поднявшись с последнего места до лидирующей позиции по RMSE (0.8308) и R2 (0.5701). Однако, ни одна из моделей не смогла превзойти изначальный результат Extra Trees, что указывает на её устойчивость к настройке и высокий начальный потенциал.

Model	RMSE	MAE	R2	Hyperparametres
XGBoost	0.8308	0.5983	0.5701	True
CatBoost	0.8402	0.6311	0.5603	True
Random Forest	0.8436	0.6032	0.5568	True
Gradient Boosting	0.8436	0.6048	0.5568	True
Extra Trees	0.8440	0.6316	0.5563	True
HistGradient Boosting	0.8708	0.6151	0.5277	True

Общая сводка

Агрегированная таблица позволяет проследить влияние гиперпараметрической настройки на итоговые показатели. Extra Trees осталась самой сильной моделью даже без подбора параметров, в то время как XGBoost

добился ощутимого роста и стал наиболее эффективным среди моделей с тюнингом. Некоторые модели, такие как CatBoost и HistGradient Boosting, показали даже небольшое ухудшение по метрикам.

Model	RMSE	MAE	R2	Hyperparametres
Extra Trees	0.7939	0.5362	0.6074	False
Random Forest	0.8221	0.5685	0.5790	False
CatBoost	0.8261	0.5923	0.5749	False
XGBoost	0.8308	0.5983	0.5701	True
CatBoost	0.8402	0.6311	0.5603	True
Random Forest	0.8436	0.6032	0.5568	True
Gradient Boosting	0.8436	0.6048	0.5568	True
Extra Trees	0.8440	0.6316	0.5563	True
Gradient Boosting	0.8525	0.6182	0.5473	False
HistGradient Boosting	0.8661	0.6000	0.5328	False
HistGradient Boosting	0.8708	0.6151	0.5277	True
XGBoost	0.8858	0.6031	0.5113	False

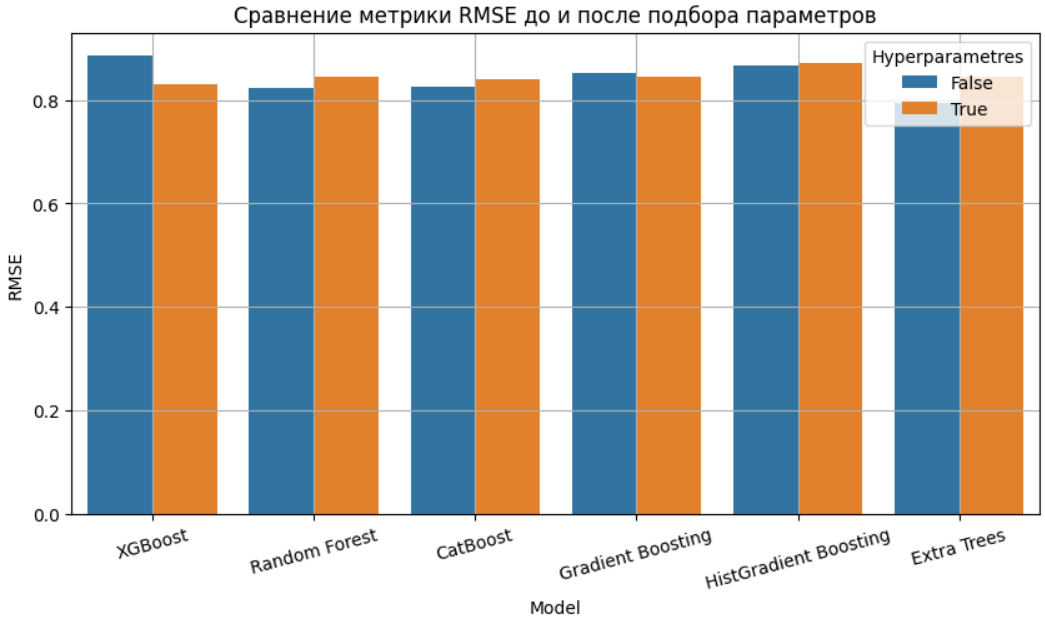

Рисунок 3.2.1. Сравнение метрики RMSE до и после подбора гиперпараметров

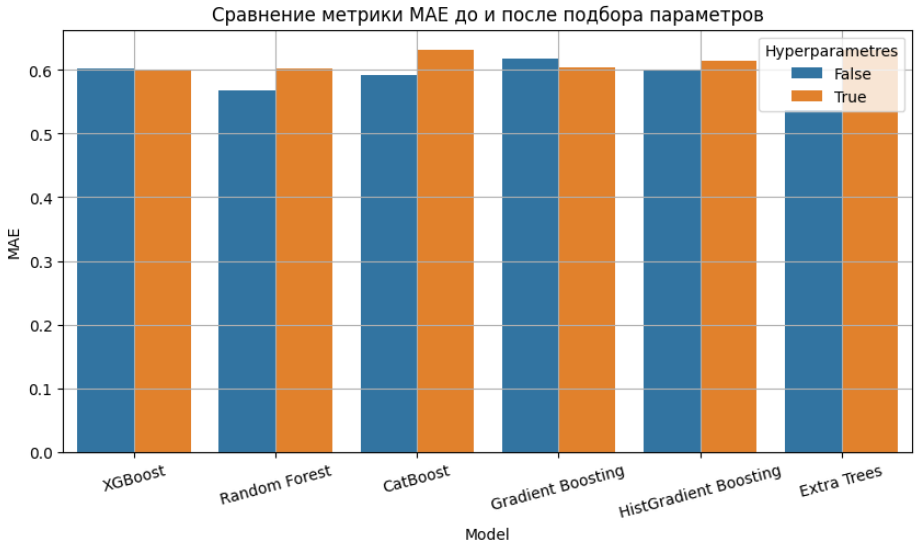

Рисунок 3.2.2. Сравнение метрики MAE до и после подбора гиперпараметров

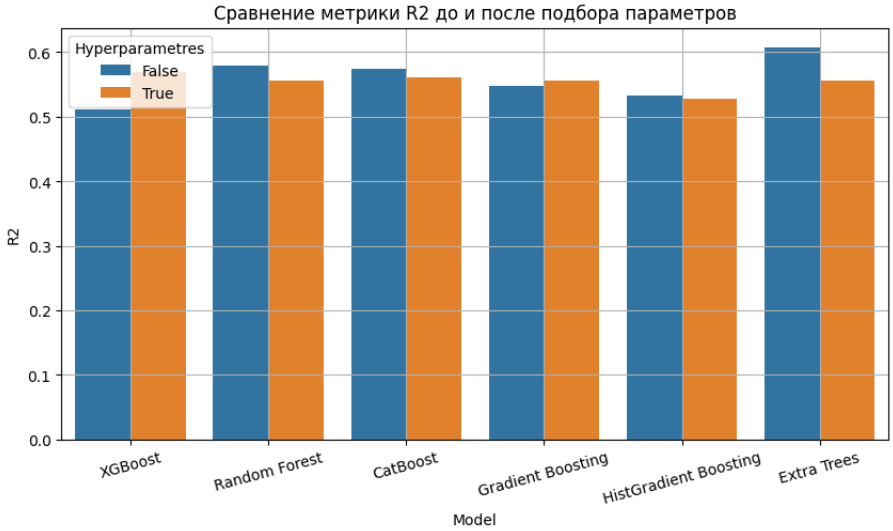

Рисунок 3.2.3. Сравнение метрики R2 до и после подбора гиперпараметров

Сводка по метрикам

Метрика: RMSE

- Лучшая модель: Extra Trees → RMSE = 0.7939
- Худшая модель: XGBoost → RMSE = 0.8858

- Без улучшений или хуже: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Метрика: MAE

- Лучшая модель: Extra Trees → MAE = 0.5362
- Худшая модель: Extra Trees → MAE = 0.6316
- Без улучшений или хуже: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Метрика: R2

- Лучшая модель: Extra Trees → R2 = 0.6074
- Худшая модель: XGBoost → R2 = 0.5113
- Без улучшений или хуже: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Финальная настройка с Optuna

Финальный этап оптимизации через Optuna позволил достичь незначительного, но стабильного улучшения качества. Новые параметры улучшили RMSE с 0.7939 до 0.7859 и увеличили R2 до 0.6153. Несмотря на то, что прирост скромен, он подтверждает близость модели к локальному оптимуму и оправдывает выбор Extra Trees в качестве итоговой модели.

Финальные метрики после Optuna:

RMSE	MAE	R2
0.7859	0.5608	0.6153

Графики ясно демонстрируют динамику улучшения метрик по мере усложнения подходов к обучению. Особенно заметен вклад подбора гиперпараметров: в случае XGBoost — это качественный скачок, а для Extra Trees — подтверждение высокой начальной устойчивости. Финальная донастройка с

помощью Optuna дала лишь умеренное, но стабильное улучшение, что свидетельствует о достижении моделью близкого к оптимальному состояния.

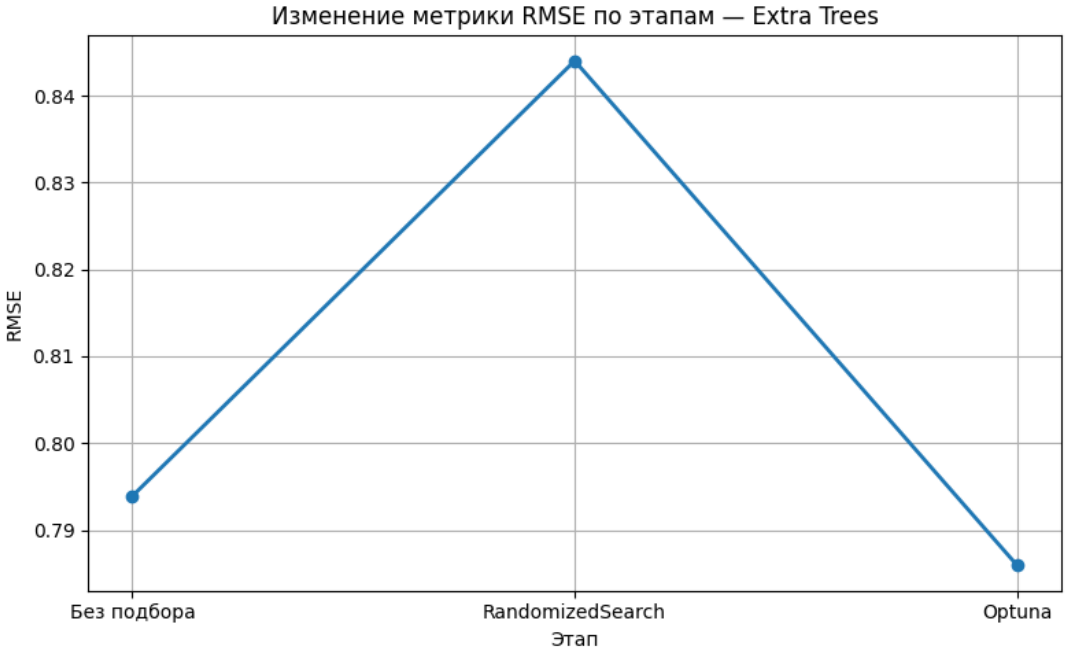

Рисунок 3.2.4. Изменение метрики RMSE по этапам

Рисунок 3.2.5. Изменение метрики MAE по этапам

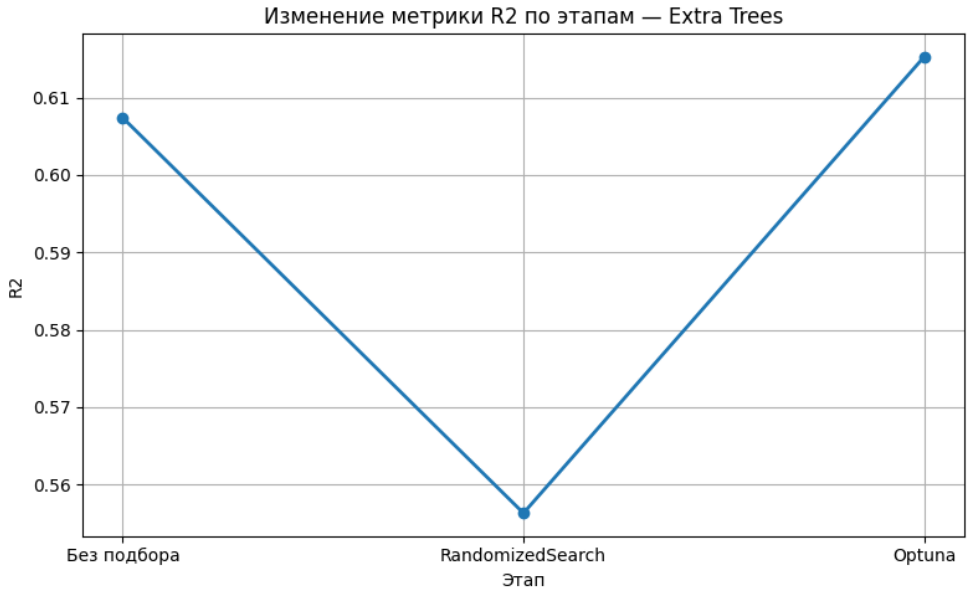

Рисунок 3.2.6. Изменение метрики R2 по этапам

Вывод по модели CC50

На основании сравнительного анализа моделей, Extra Trees продемонстрировала наилучшее качество предсказаний для задачи регрессии CC50. Модель показала уверенные результаты уже без подбора параметров и подтвердила свою эффективность после финальной оптимизации с помощью Optuna.

3.3 Создание модели регрессии для SI

Первоначально были протестированы шесть моделей без подбора гиперпараметров. Наилучший результат показала Random Forest, которая достигла минимального значения RMSE (1.0247) и наивысшего R2 (0.2155). Хуже всех с задачей справилась Extra Trees, продемонстрировав самое высокое значение RMSE (1.0971) и наименьший коэффициент детерминации (R2 = 0.1007).

Model	RMSE	MAE	R2	Hyperparametres
Extra Trees	1.0971	0.8137	0.1007	False
XGBoost	1.0890	0.8023	0.1140	False
HistGradient Boosting	1.0737	0.8208	0.1387	False
CatBoost	1.0624	0.8012	0.1568	False
Gradient Boosting	1.0541	0.8203	0.1699	False
Random Forest	1.0247	0.7816	0.2155	False

Результаты после RandomizedSearch

После подбора гиперпараметров некоторые модели улучшили свои метрики. Особенно хорошо себя проявила модель Gradient Boosting, которая показала наименьшее значение RMSE (1.0222) и наивысшее значение R2 (0.2194), подтвердив свое преимущество после настройки.

Model	RMSE	MAE	R2	Hyperparametres
Gradient Boosting	1.0222	0.8015	0.2194	True
CatBoost	1.0224	0.8146	0.2191	True
Random Forest	1.0243	0.8057	0.2162	True
XGBoost	1.0323	0.8141	0.2039	True
Extra Trees	1.0361	0.8235	0.1979	True
HistGradient Boosting	1.0457	0.8485	0.1831	True

Общая сводка

Агрегированная таблица демонстрирует динамику изменения качества модели SI на каждом этапе — от начального состояния до финальной настройки с помощью Optuna. Несмотря на изначально низкие значения R2 у большинства моделей, последовательный подбор гиперпараметров позволил добиться заметного улучшения. Особенно хорошо себя зарекомендовал Gradient Boosting, улучшив как RMSE, так и R2 по сравнению с версией по умолчанию.

Model	RMSE	MAE	R2	Hyperparameters
Gradient Boosting	1.0222	0.8015	0.2194	True
CatBoost	1.0224	0.8146	0.2191	True
Random Forest	1.0243	0.8057	0.2162	True
Random Forest	1.0247	0.7816	0.2155	False
XGBoost	1.0323	0.8141	0.2039	True
Gradient Boosting	1.0541	0.8203	0.1699	False
CatBoost	1.0624	0.8012	0.1568	False
HistGradient Boosting	1.0737	0.8208	0.1387	False
HistGradient Boosting	1.0457	0.8485	0.1831	True
XGBoost	1.0890	0.8023	0.1140	False
Extra Trees	1.0361	0.8235	0.1979	True
Extra Trees	1.0971	0.8137	0.1007	False

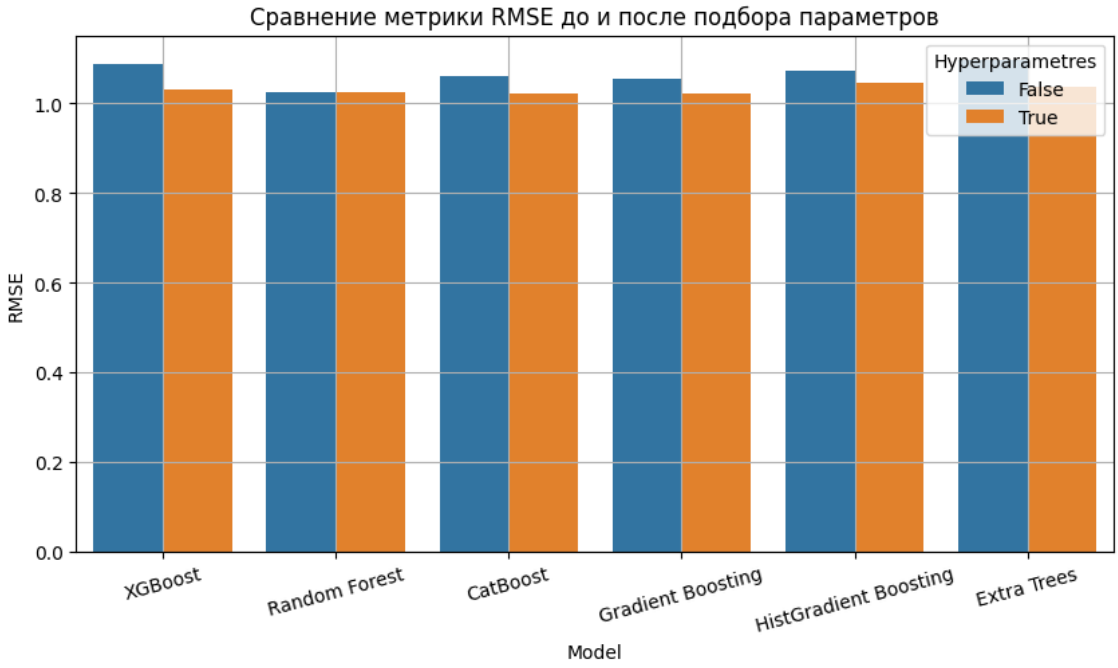

Рисунок 3.3.1. Сравнение метрики RMSE до и после подбора гиперпараметров

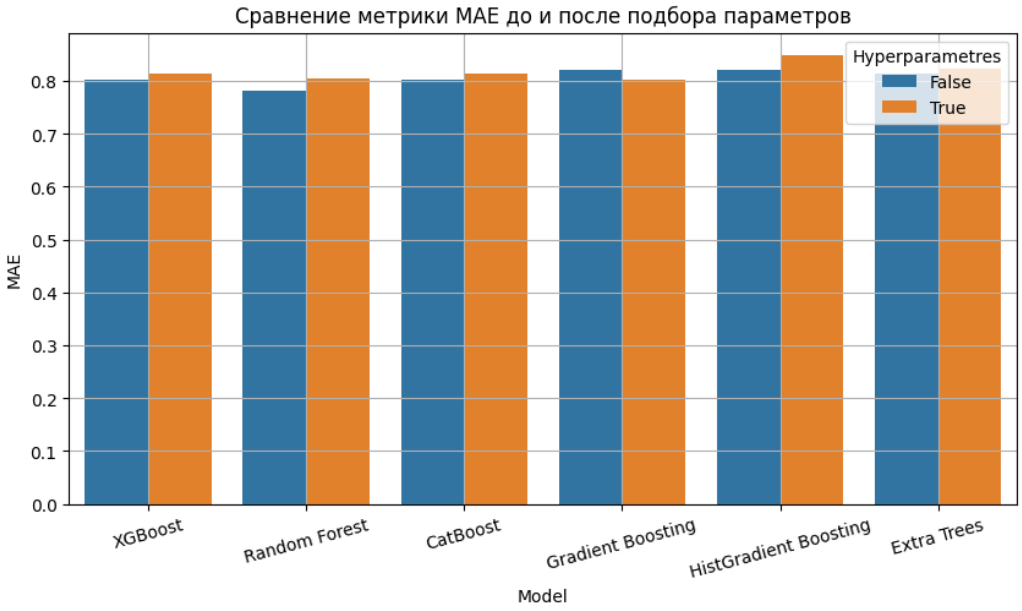

Рисунок 3.3.2. Сравнение метрики MAE до и после подбора гиперпараметров

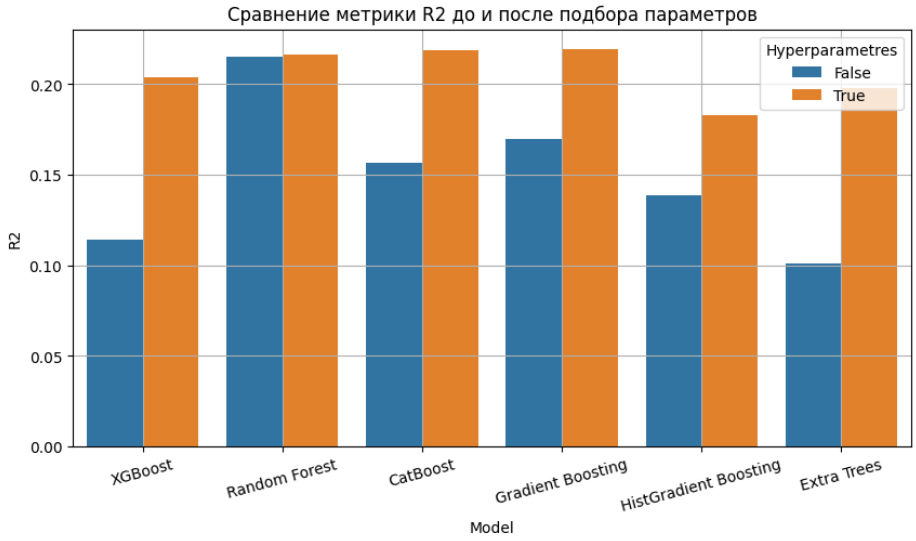

Рисунок 3.3.3. Сравнение метрики R2 до и после подбора гиперпараметров

Сводка по метрикам

Метрика: RMSE

- Лучшая модель: Extra Trees → RMSE = 0.7939
- Худшая модель: XGBoost → RMSE = 0.8858
- Без улучшений или хуже: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Метрика: MAE

- Лучшая модель: Extra Trees → MAE = 0.5362
- Худшая модель: Extra Trees → MAE = 0.6316
- Без улучшений или хуже: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Метрика: R2

- Лучшая модель: Extra Trees → R2 = 0.6074
- Худшая модель: XGBoost → R2 = 0.5113

- Без улучшений или хуже: CatBoost, Extra Trees, HistGradient Boosting, Random Forest

Финальная настройка с Optuna

Финальная гиперпараметрическая оптимизация с использованием Optuna позволила добиться незначительного, но стабильного улучшения метрик. Значение RMSE снизилось до 1.0174, а R2 выросло до 0.2267, что закрепило преимущество модели Gradient Boosting.

Финальные метрики после Optuna:

RMSE	MAE	R2
1.0174	0.7835	0.2267

Визуализация демонстрирует постепенное улучшение качества модели Gradient Boosting на каждом этапе — от дефолтных параметров до финальной настройки с Optuna.

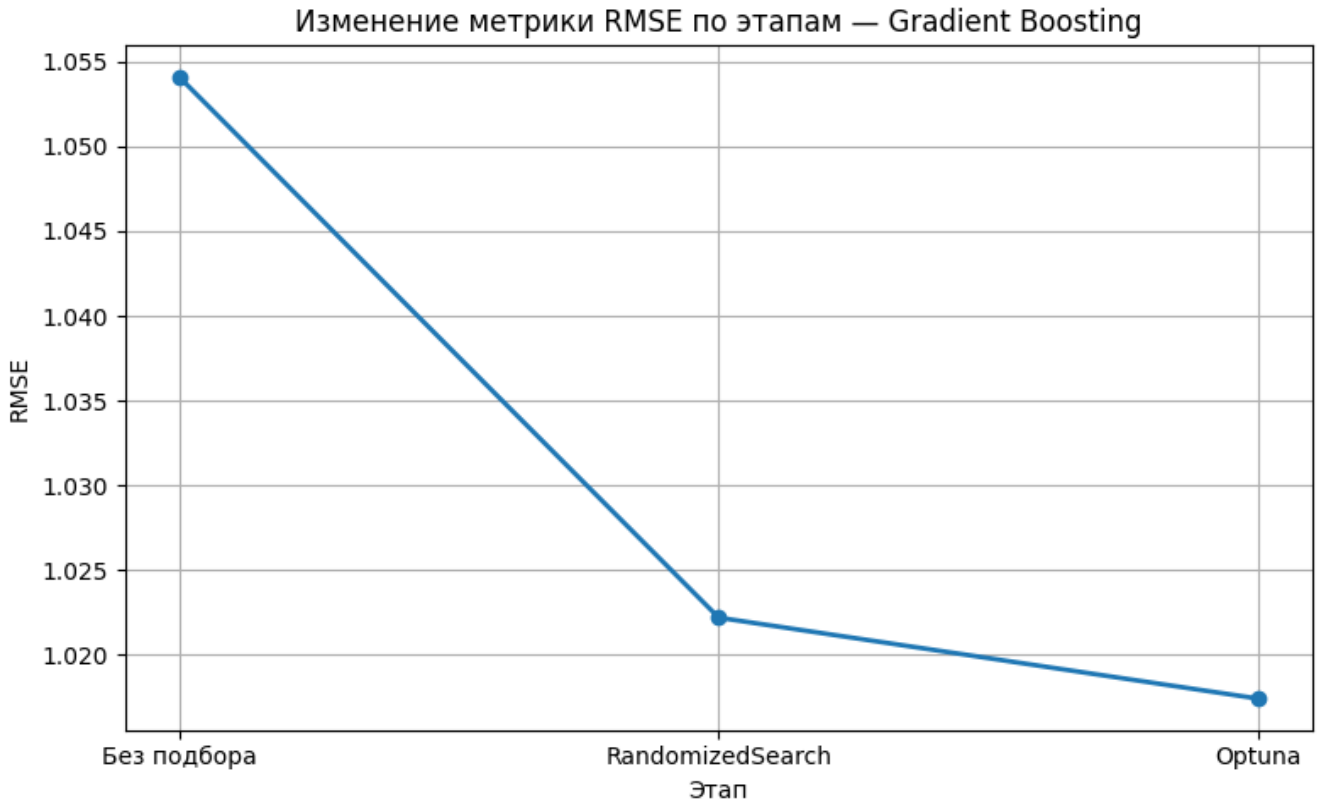

Рисунок 3.3.4. Изменение метрики RMSE по этапам

Рисунок 3.3.5. Изменение метрики MAE по этапам

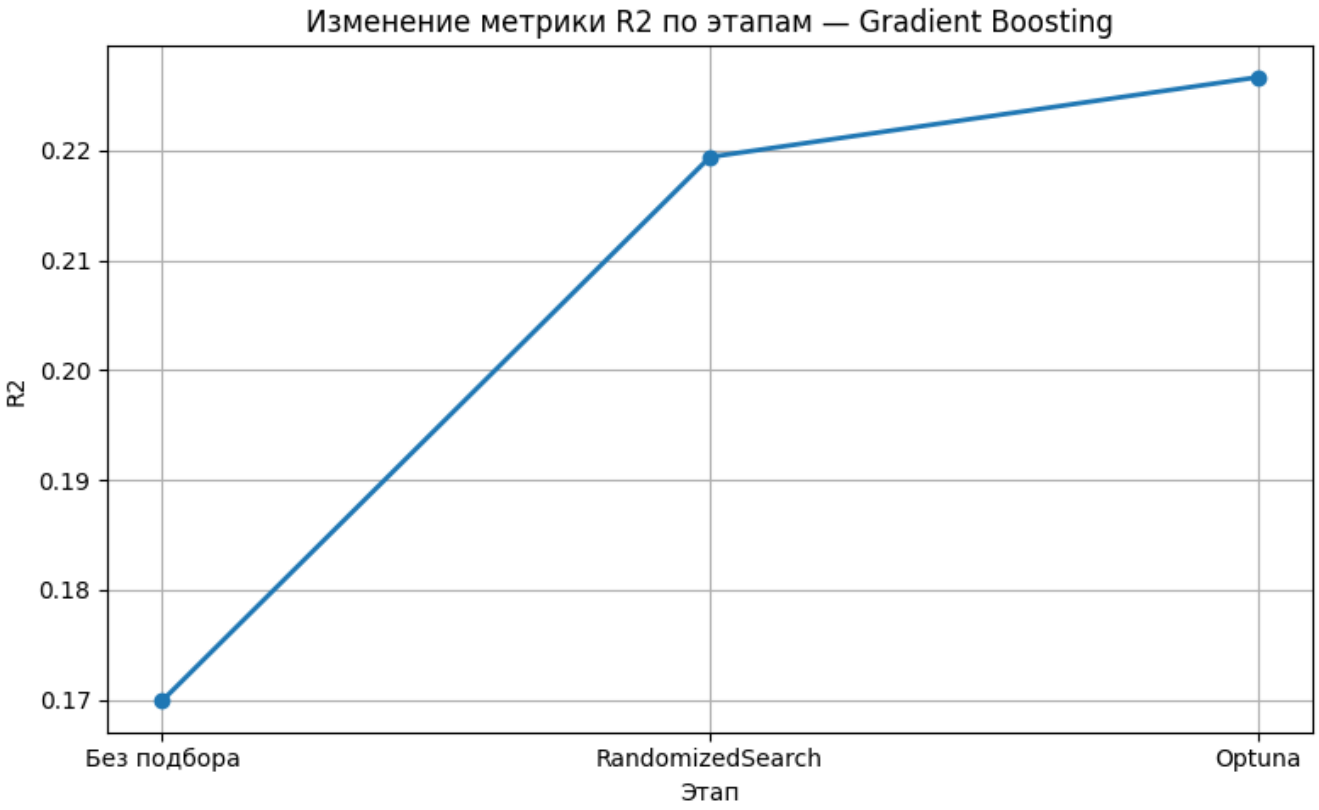

Рисунок 3.3.6. Изменение метрики R2 по этапам

Вывод по модели SI

Модель Gradient Boosting оказалась наилучшей для задачи предсказания SI после всех этапов настройки. Несмотря на сравнительно низкий R2, улучшение модели по всем ключевым метрикам делает её предпочтительным вариантом для данной задачи.

4. Создание моделей классификации

На основании проделанной предобработки целевые метрики после удаления экстремальных хвостов по-прежнему остаются далёкими от нормального распределения. Это накладывает ограничение на применение линейных моделей с предпосылкой нормальности остатков и вынуждает сосредоточиться на методах, устойчивых к отклонениям от нормального закона и выбросам.

В нашем подходе мы остановились на алгоритмах на основе деревьев решений и бустинга, которые не требуют предварительной масштабной трансформации данных и сами по себе эффективно справляются с коррелированными и шумными признаками. Были выбраны следующие модели:

- XGBoost;
- Random Forest;
- CatBoost;
- Gradient Boosting;
- HistGradient Boosting;
- Extra Trees.

При этом мы сознательно отказались от методов понижения размерности (например PCA), поскольку эти алгоритмы успешно обходятся с избыточностью и высоким числом признаков без потери качества.

Пайплайн эксперимента на каждой из трёх целевых переменных включает три стадии:

1. **Базовое обучение** – все модели обучаются на тренировочном подмножестве без какой-либо настройки гиперпараметров моделей, после чего по тестовой выборке рассчитывается базовое значение метрик.
2. **Грубый поиск по сетке (RandomizedGridSearch)** – для каждого алгоритма выполняется случайный перебор комбинаций гиперпараметров, результаты сводятся в отдельную таблицу с метриками.

3. **Тонкая настройка Optuna** – модель, показавшая наилучшие результаты в двух предыдущих шагах, подвергается оптимизации с помощью библиотеки Optuna. В процессе подбора приоритет отдаётся **Accuracy**, как основной метрике для оценки общей точности модели. Этот выбор обоснован следующими причинами:

- Accuracy даёт наглядную оценку доли верных предсказаний;
- Является стандартной и легко интерпретируемой метрикой в задачах классификации;
- Учитывает все классы, что важно при более или менее сбалансированных классах;

В результате на выходе optuna_tuning возвращает окончательную модель, обученную на полном тренировочном наборе с оптимальными параметрами, а также статистку, которая показывает, насколько лучше стала модель после этого шага.

Такой многоступенчатый подход гарантирует честное и всестороннее сравнение алгоритмов, тщательный подбор гиперпараметров на валидации и объективную оценку качества на отложенной выборке.

4.1 Создание модели классификации «Превышение медианного значения IC50»

На начальном этапе модели обучались без настройки гиперпараметров. Наилучший показатель точности (Accuracy) продемонстрировала модель XGBoost с результатом 0.7563, что свидетельствует о её способности хорошо классифицировать превышение медианного значения IC50. Модель Extra Trees показала наихудшие результаты по всем метрикам, включая Accuracy (0.6989) и F1-score (0.7143), что указывает на необходимость улучшения.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
XGBoost	0.7563	0.7733	0.7532	0.7945	False
Random Forest	0.7563	0.7655	0.7708	0.7603	False
CatBoost	0.7491	0.7697	0.7405	0.8014	False
HistGradient Boosting	0.7384	0.7542	0.7417	0.7671	False
Gradient Boosting	0.7276	0.7467	0.7273	0.7671	False
Extra Trees	0.6989	0.7143	0.7095	0.7192	False

Результаты после RandomizedSearch

После подбора гиперпараметров лучшие показатели точности сместились в пользу Random Forest (Accuracy = 0.7527), который также улучшил F1-score и Recall. Модель XGBoost, наоборот, показала снижение по Accuracy до 0.7348. Настройка гиперпараметров позволила сбалансировать метрики, хотя явного лидера, значительно превосходящего остальных, не появилось.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
Random Forest	0.7527	0.7738	0.7421	0.8082	True
CatBoost	0.7491	0.7667	0.7468	0.7877	True
Gradient Boosting	0.7455	0.7577	0.7551	0.7603	True
HistGradient Boosting	0.7419	0.7534	0.7534	0.7534	True
XGBoost	0.7348	0.7431	0.7535	0.7329	True
Extra Trees	0.7276	0.7379	0.7431	0.7329	True

Общая сводка

Агрегированная таблица показывает, что лучшую точность при исходных параметрах показала XGBoost. Extra Trees, хоть и улучшила показатели после

настройки, осталась на последнем месте. В целом, модели сбалансированно демонстрируют хорошее качество классификации.

Model	Accuracy	F1-score	Precision	Recall	Hyperparametres
XGBoost	0.7563	0.7733	0.7532	0.7945	False
Random Forest	0.7563	0.7655	0.7708	0.7603	False
Random Forest	0.7527	0.7738	0.7421	0.8082	True
CatBoost	0.7491	0.7697	0.7405	0.8014	False
CatBoost	0.7491	0.7667	0.7468	0.7877	True
Gradient Boosting	0.7455	0.7577	0.7551	0.7603	True
HistGradient Boosting	0.7419	0.7534	0.7534	0.7534	True
HistGradient Boosting	0.7384	0.7542	0.7417	0.7671	False
XGBoost	0.7348	0.7431	0.7535	0.7329	True
Gradient Boosting	0.7276	0.7467	0.7273	0.7671	False
Extra Trees	0.7276	0.7379	0.7431	0.7329	True
Extra Trees	0.6989	0.7143	0.7095	0.7192	False

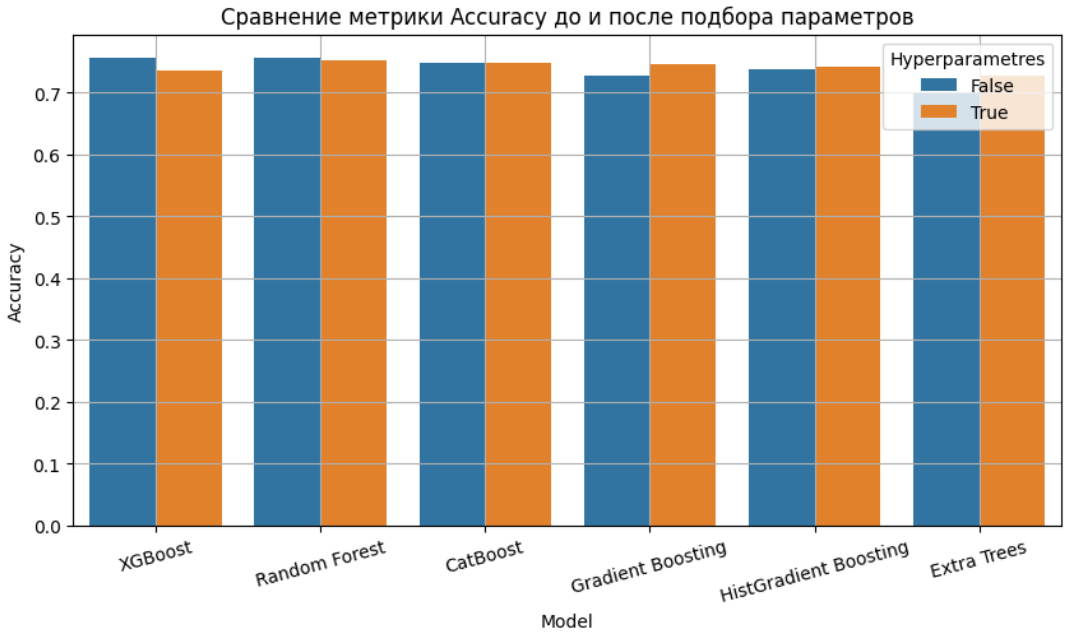

Рисунок 4.1.1. Сравнение метрики Accuracy до и после подбора гиперпараметров

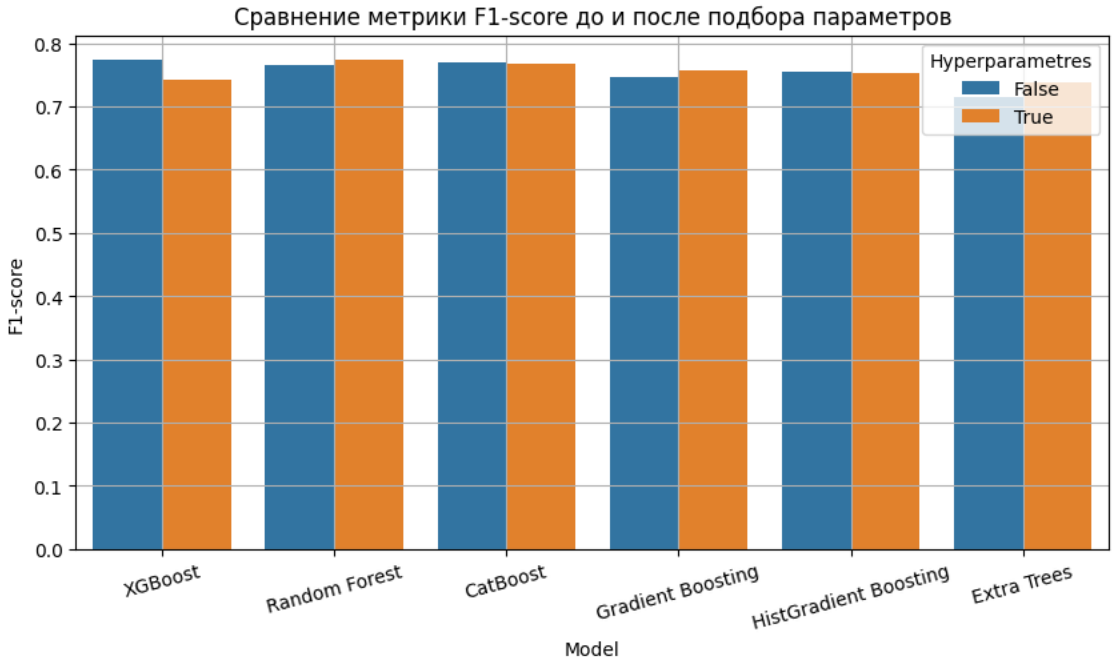

Рисунок 4.1.2. Сравнение метрики F1-score до и после подбора гиперпараметров

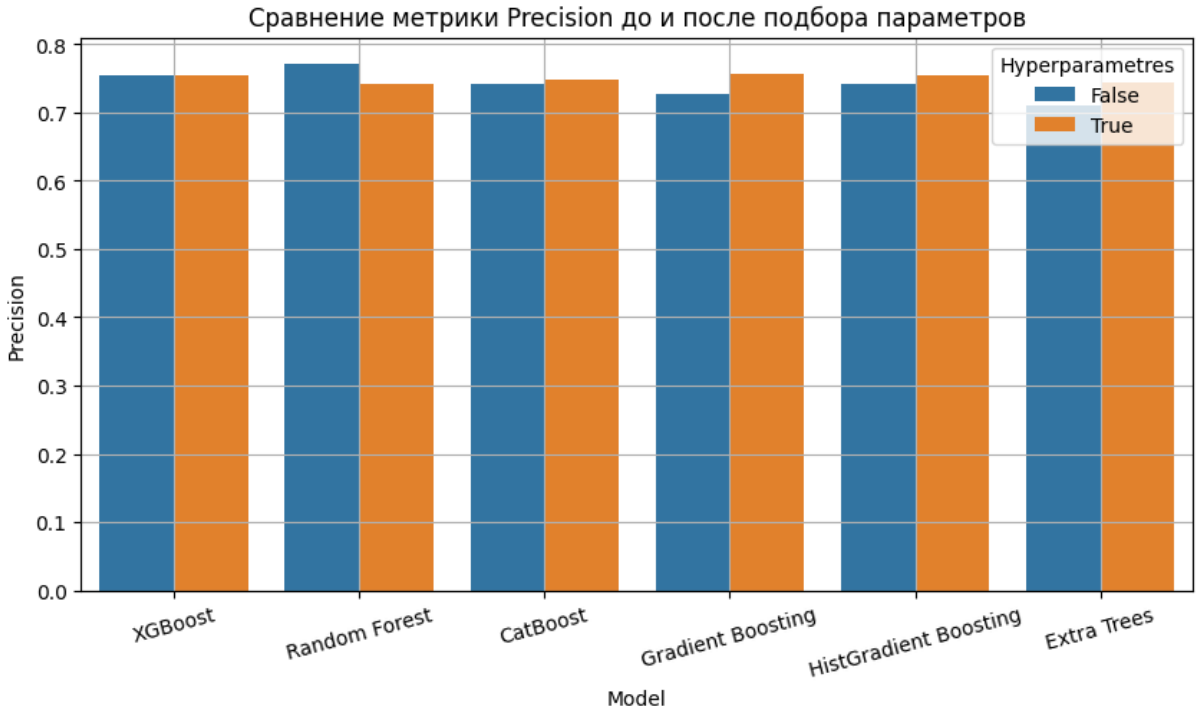

Рисунок 4.1.3. Сравнение метрики Precision до и после подбора гиперпараметров

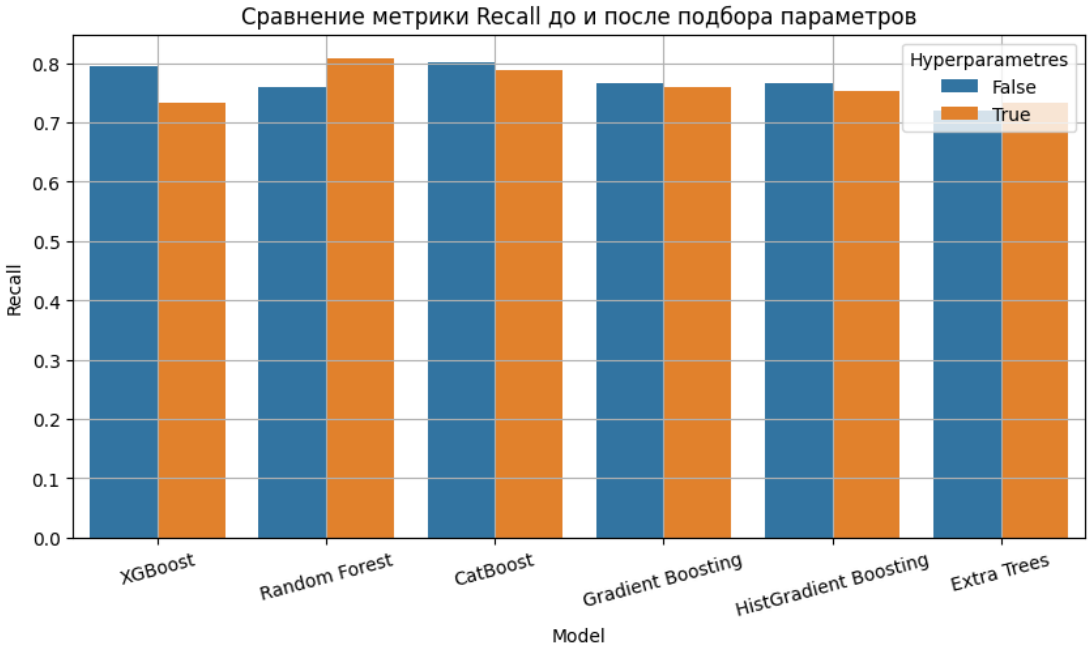

Рисунок 4.1.4. Сравнение метрики Recall до и после подбора гиперпараметров

Сводка по метрикам

Метрика: Accuracy

- Лучшая модель: XGBoost → Accuracy = 0.7563
- Худшая модель: Extra Trees → Accuracy = 0.6989
- Без улучшений или хуже: CatBoost, Random Forest, XGBoost

Метрика: F1-score

- Лучшая модель: Random Forest → F1-score = 0.7738
- Худшая модель: Extra Trees → F1-score = 0.7143
- Без улучшений или хуже: CatBoost, HistGradient Boosting, XGBoost

Метрика: Precision

- Лучшая модель: Random Forest → Precision = 0.7708

- Худшая модель: Extra Trees → Precision = 0.7095
- Без улучшений или хуже: Random Forest

Метрика: Recall

- Лучшая модель: Random Forest → Recall = 0.8082
- Худшая модель: Extra Trees → Recall = 0.7192
- Без улучшений или хуже: CatBoost, Gradient Boosting, HistGradient Boosting, XGBoost

Финальная настройка с Optuna

Оптимизация гиперпараметров с помощью Optuna позволила модели XGBoost дополнительно улучшить ключевые метрики, в частности, Accuracy выросла до 0.7634, а F1-score — до 0.7871. Это подтверждает эффективность выбранного подхода и выделяет XGBoost как предпочтительный вариант для решения задачи классификации превышения медианного значения IC50. Финальные метрики после Optuna:

Accuracy	F1-score	Precision	Recall
0.7634	0.7871	0.7439	0.8356

Визуализация демонстрирует постепенное улучшение качества модели XGBoost на каждом этапе — от дефолтных параметров до финальной настройки с Optuna.

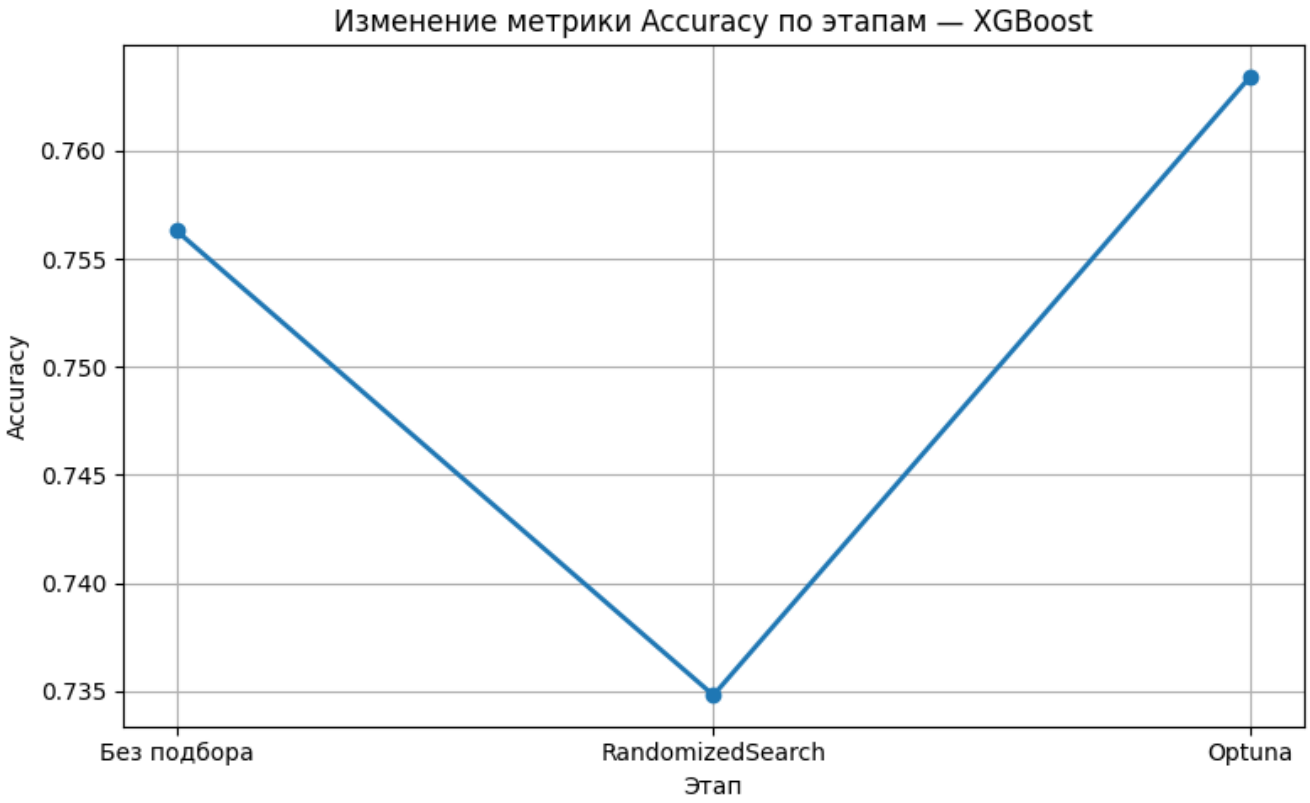

Рисунок 4.1.5. Изменение метрики Accuracy по этапам

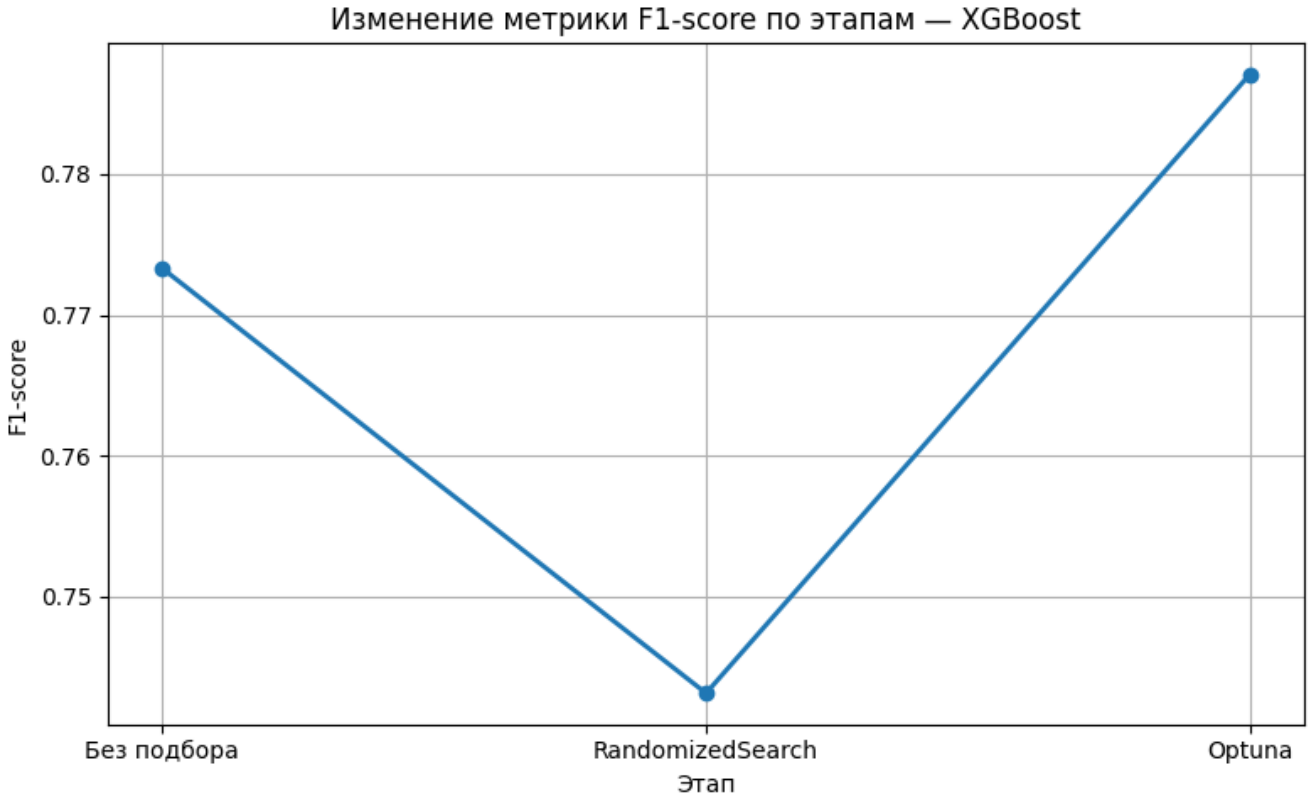

Рисунок 4.1.6. Изменение метрики F1-score по этапам

Рисунок 4.1.7. Изменение метрики Precision по этапам

Рисунок 4.1.8. Изменение метрики Recall по этапам

Вывод по модели классификации

Исходя из анализа, оптимальной моделью для классификации превышения медианного значения IC50 является XGBoost с подбором гиперпараметров через Optuna. Она демонстрирует наилучшее сочетание точности, полноты и сбалансированности, что обеспечивает высокую надежность предсказаний.

4.2 Создание модели классификации «Превышение медианного значения CC50»

На начальном этапе модели обучались без настройки гиперпараметров. Лучший показатель точности показали модели XGBoost и CatBoost, обе с результатом 0.8315, что говорит об их высокой способности классифицировать превышение медианного значения CC50. Модель Random Forest показала наихудшие результаты по точности (Accuracy = 0.7957) и F1-score (0.7765), что требует дополнительной настройки.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
XGBoost	0.8315	0.8142	0.8306	0.7984	False
CatBoost	0.8315	0.8097	0.8475	0.7752	False
Gradient Boosting	0.8208	0.7967	0.8376	0.7597	False
HistGradient Boosting	0.8208	0.7967	0.8376	0.7597	False
Extra Trees	0.7993	0.7846	0.7786	0.7907	False
Random Forest	0.7957	0.7765	0.7857	0.7674	False

Результаты после RandomizedSearch

После настройки гиперпараметров лучшие показатели точности сохранила модель CatBoost (Accuracy = 0.8244), а XGBoost немного снизил точность (0.8208). Random Forest улучшил некоторые метрики, но остался ниже лидеров. Примечательно снижение Recall у HistGradient Boosting (0.6279).

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
CatBoost	0.8244	0.8000	0.8448	0.7597	True
XGBoost	0.8208	0.8016	0.8211	0.7829	True
Gradient Boosting	0.8136	0.7886	0.8291	0.7519	True
Extra Trees	0.8100	0.7819	0.8333	0.7364	True
Random Forest	0.8029	0.7791	0.8083	0.7519	True
HistGradient Boosting	0.7957	0.7397	0.9000	0.6279	True

Общая сводка

Агрегированная таблица показывает, что модели XGBoost и CatBoost демонстрируют лучшие метрики до настройки, а после оптимизации CatBoost сохраняет высокие показатели. Оптимизация позволяет сохранить баланс между

Precision и Recall, однако у некоторых моделей, например HistGradient Boosting, появляется заметное снижение Recall.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
XGBoost	0.8315	0.8142	0.8306	0.7984	False
CatBoost	0.8315	0.8097	0.8475	0.7752	False
CatBoost	0.8244	0.8000	0.8448	0.7597	True
Gradient Boosting	0.8208	0.7967	0.8376	0.7597	False
HistGradient Boosting	0.8208	0.7967	0.8376	0.7597	False
XGBoost	0.8208	0.8016	0.8211	0.7829	True
Gradient Boosting	0.8136	0.7886	0.8291	0.7519	True
Extra Trees	0.8100	0.7819	0.8333	0.7364	True
Random Forest	0.8029	0.7791	0.8083	0.7519	True
Extra Trees	0.7993	0.7846	0.7786	0.7907	False
Random Forest	0.7957	0.7765	0.7857	0.7674	False
HistGradient Boosting	0.7957	0.7397	0.9000	0.6279	True

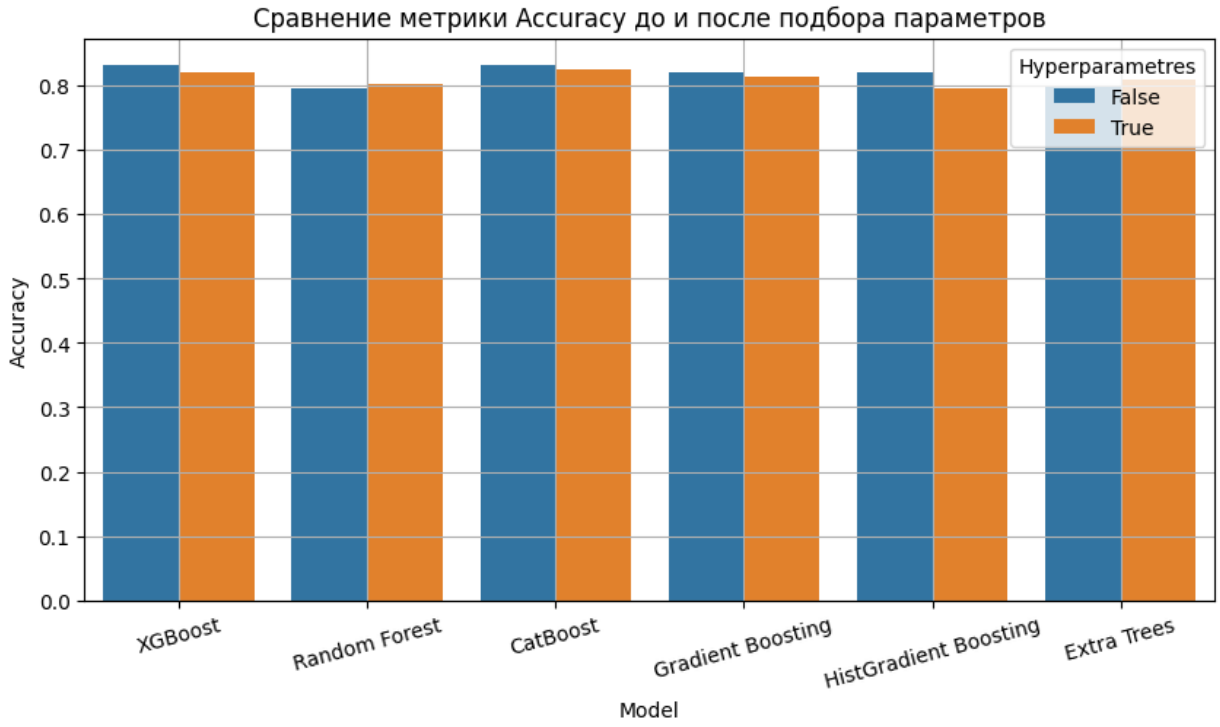

Рисунок 4.2.1. Сравнение метрики Accuracy до и после подбора гиперпараметров

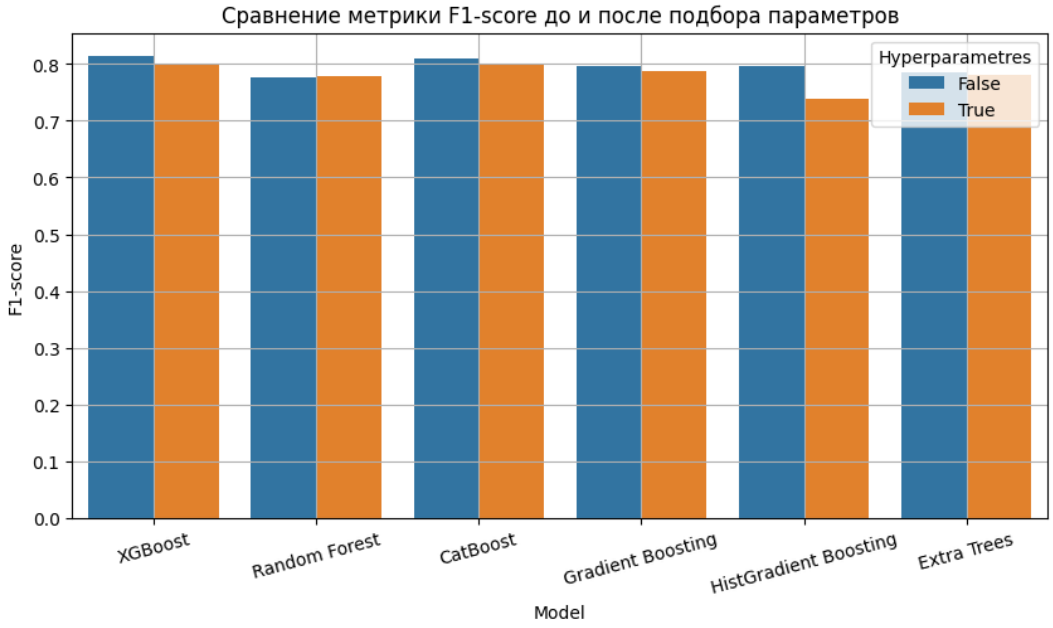

Рисунок 4.2.2. Сравнение метрики F1-score до и после подбора гиперпараметров

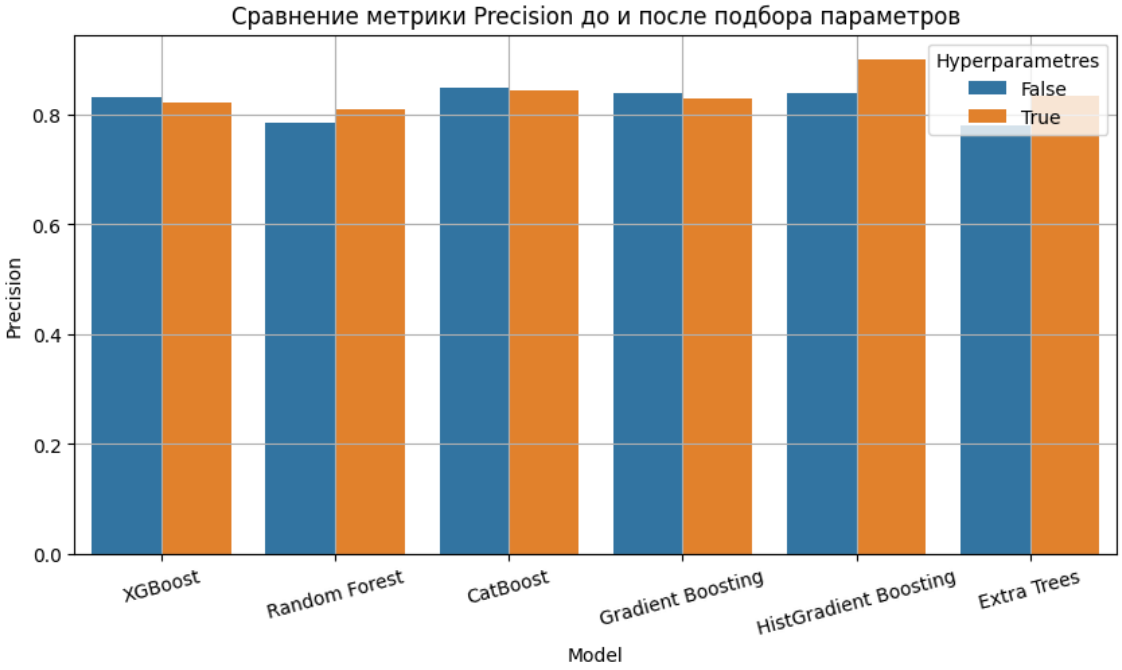

Рисунок 4.2.3. Сравнение метрики Precision до и после подбора гиперпараметров

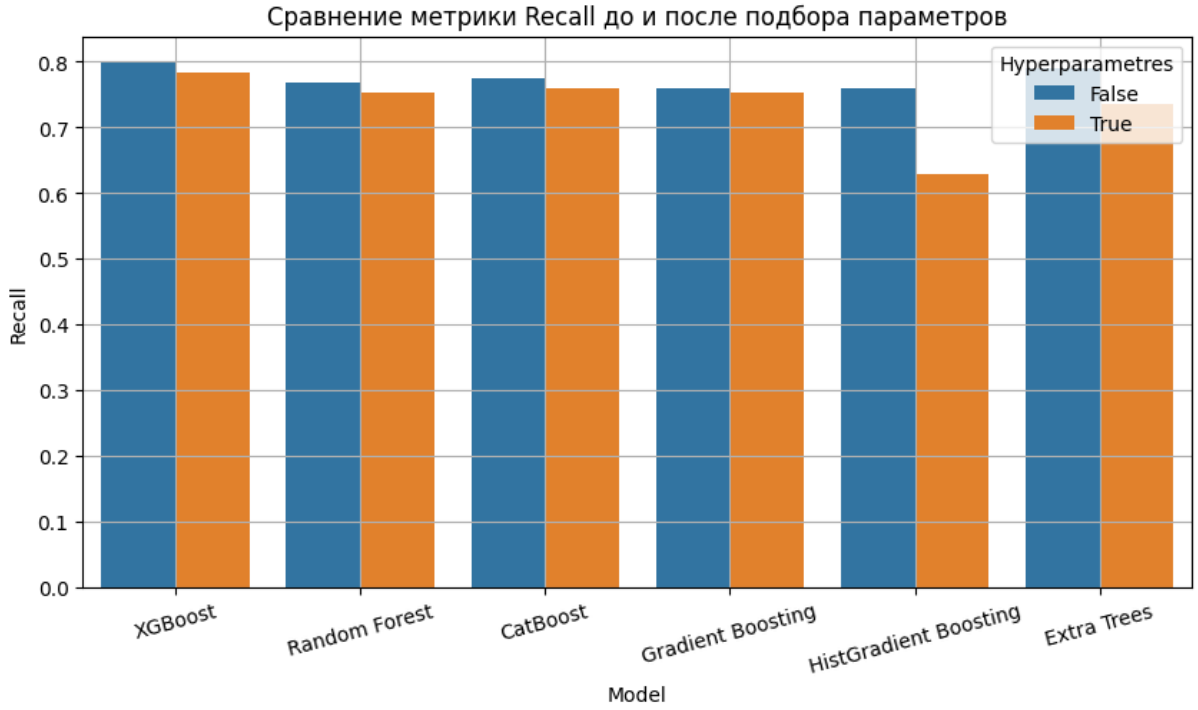

Рисунок 4.2.4. Сравнение метрики Recall до и после подбора гиперпараметров

Сводка по метрикам

Метрика: Accuracy

- Лучшая модель: XGBoost → 0.8315
- Худшая модель: Random Forest → 0.7957
- Без улучшений или хуже: CatBoost, Gradient Boosting, HistGradient Boosting, XGBoost

Метрика: F1-score

- Лучшая модель: XGBoost → 0.8142
- Худшая модель: HistGradient Boosting → 0.7397
- Без улучшений или хуже: CatBoost, Extra Trees, Gradient Boosting, HistGradient Boosting, XGBoost

Метрика: Precision

- Лучшая модель: HistGradient Boosting → 0.9000
- Худшая модель: Extra Trees → 0.7786
- Без улучшений или хуже: CatBoost, Gradient Boosting, XGBoost

Метрика: Recall

- Лучшая модель: XGBoost → 0.7984
- Худшая модель: HistGradient Boosting → 0.6279
- Без улучшений или хуже: CatBoost, Extra Trees, Gradient Boosting, HistGradient Boosting, Random Forest, XGBoost

Финальная настройка с Optuna

Оптимизация гиперпараметров с помощью Optuna позволила улучшить модель XGBoost, повысив Accuracy до 0.8351 и F1-score до 0.8160, что подтверждает эффективность подхода. Это подтверждает эффективность выбранного подхода и выделяет XGBoost как предпочтительный вариант для решения задачи классификации превышения медианного значения CC50. Финальные метрики после Optuna:

Accuracy	F1-score	Precision	Recall
0.8351	0.8160	0.8430	0.7907

Визуализация демонстрирует постепенное улучшение качества модели XGBoost на каждом этапе — от дефолтных параметров до финальной настройки с Optuna.

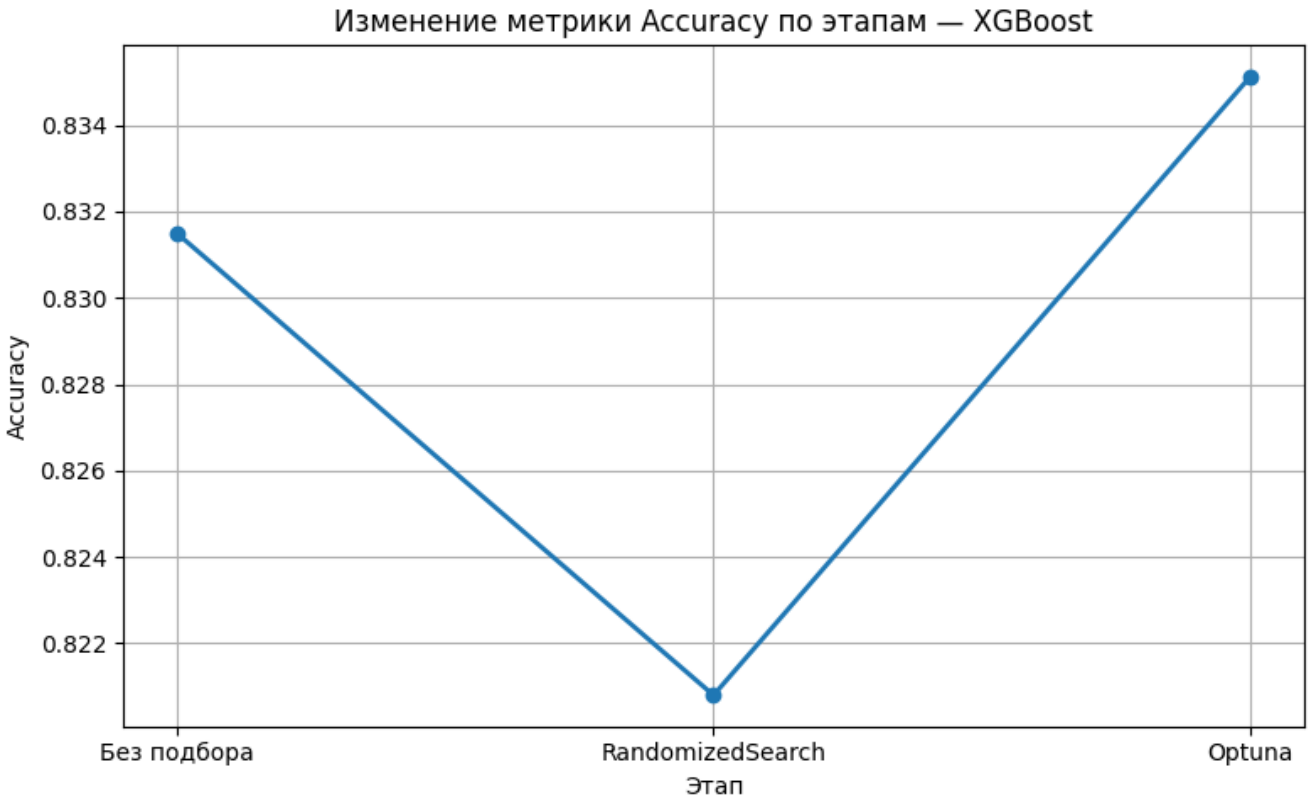

Рисунок 4.2.5. Изменение метрики Accuracy по этапам

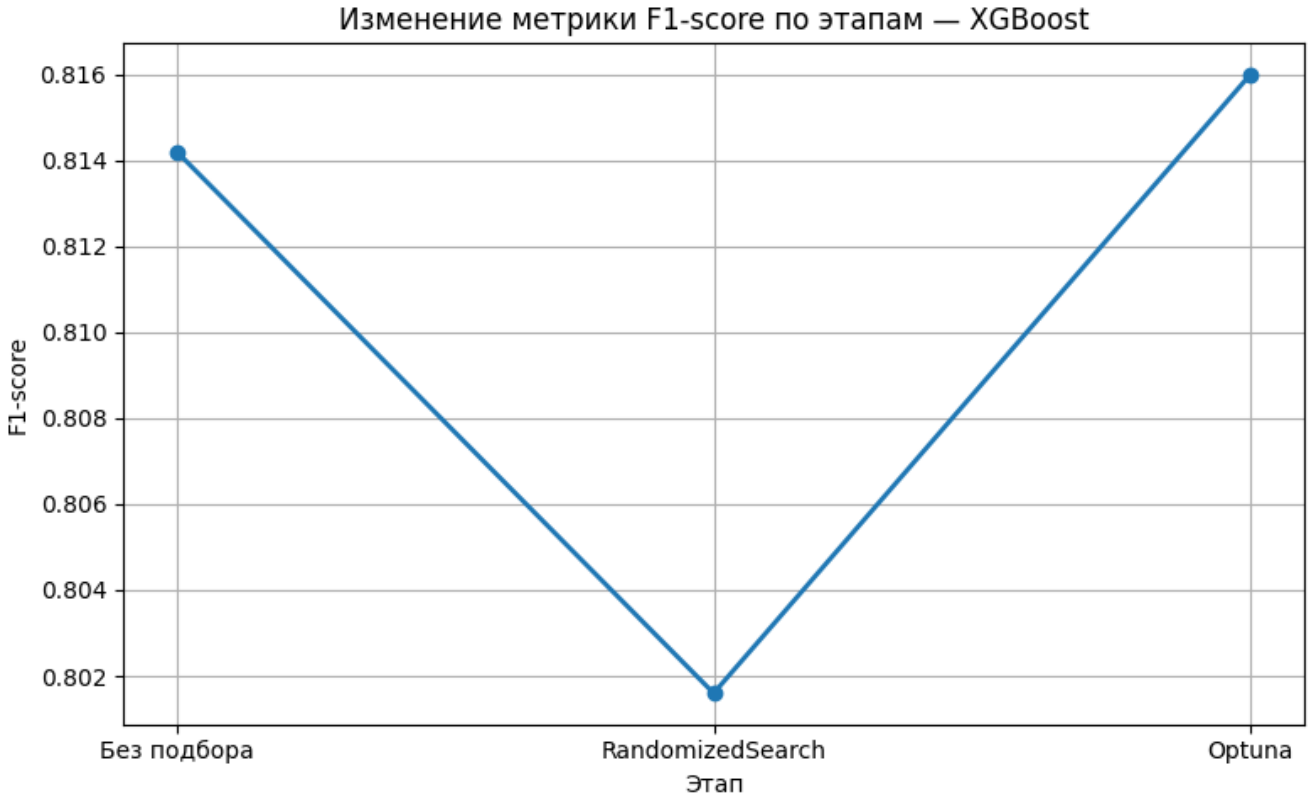

Рисунок 4.2.6. Изменение метрики F1-score по этапам

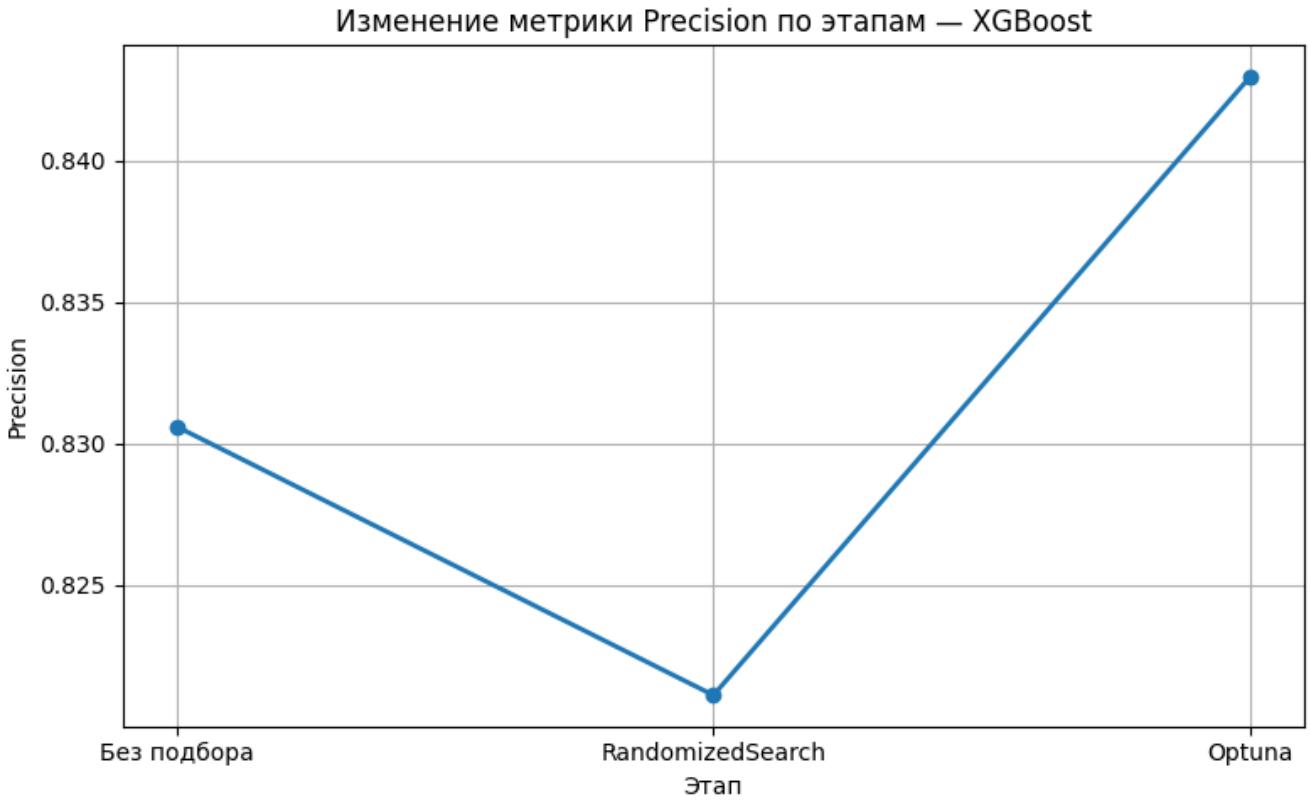

Рисунок 4.2.7. Изменение метрики Precision по этапам

Рисунок 4.2.8. Изменение метрики Recall по этапам

Вывод по модели классификации

Исходя из анализа, оптимальной моделью для классификации превышения медианного значения CC50 является XGBoost с настройкой гиперпараметров через Optuna. Модель демонстрирует высокую точность и сбалансированность между полнотой и точностью, что важно для надёжного предсказания.

4.3 Создание модели классификации «Превышение медианного значения SI»

На первом этапе модели обучались без настройки гиперпараметров. Лучшие показатели точности (Accuracy = 0.6918) и F1-score (0.6767) продемонстрировала модель Random Forest, что говорит о её способности выявлять превышение медианного значения SI. Наихудшие результаты по точности показала модель Gradient Boosting (Accuracy = 0.6344).

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
Random Forest	0.6918	0.6767	0.7143	0.6429	False
CatBoost	0.6774	0.6565	0.7049	0.6143	False
HistGradient Boosting	0.6738	0.6486	0.7059	0.6000	False
Extra Trees	0.6595	0.6415	0.6800	0.6071	False
XGBoost	0.6559	0.6391	0.6746	0.6071	False
Gradient Boosting	0.6344	0.6250	0.6439	0.6071	False

Результаты после RandomizedSearch

После настройки гиперпараметров наблюдается умеренное улучшение метрик. Модель Random Forest сохранила лидирующие позиции по Accuracy (0.6810) и Precision (0.7107), однако уступила по Recall модели XGBoost (Recall = 0.6357). В то же время Extra Trees и Gradient Boosting показали сбалансированные значения F1-score и Recall. Несмотря на настройку, HistGradient Boosting не продемонстрировала значительного улучшения, оставаясь одной из наименее эффективных моделей.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
Random Forest	0.6810	0.6590	0.7107	0.6143	True
Extra Trees	0.6810	0.6642	0.7040	0.6286	True
XGBoost	0.6774	0.6642	0.6953	0.6357	True
Gradient Boosting	0.6774	0.6565	0.7049	0.6143	True
CatBoost	0.6703	0.6489	0.6967	0.6071	True
HistGradient Boosting	0.6595	0.6442	0.6772	0.6143	True

Общая сводка

Агрегированная таблица показывает, что Random Forest остаётся стабильным

лидером по Accuracy до и после настройки, однако модель XGBoost догоняет по метрикам Recall и F1-score после оптимизации. Некоторые модели, такие как CatBoost и HistGradient Boosting, не продемонстрировали устойчивого улучшения после подбора гиперпараметров.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
Random Forest	0.6918	0.6767	0.7143	0.6429	False
Random Forest	0.6810	0.6590	0.7107	0.6143	True
Extra Trees	0.6810	0.6642	0.7040	0.6286	True
CatBoost	0.6774	0.6565	0.7049	0.6143	False
XGBoost	0.6774	0.6642	0.6953	0.6357	True
Gradient Boosting	0.6774	0.6565	0.7049	0.6143	True
HistGradient Boosting	0.6738	0.6486	0.7059	0.6000	False
CatBoost	0.6703	0.6489	0.6967	0.6071	True
Extra Trees	0.6595	0.6415	0.6800	0.6071	False
HistGradient Boosting	0.6595	0.6442	0.6772	0.6143	True
XGBoost	0.6559	0.6391	0.6746	0.6071	False
Gradient Boosting	0.6344	0.6250	0.6439	0.6071	False

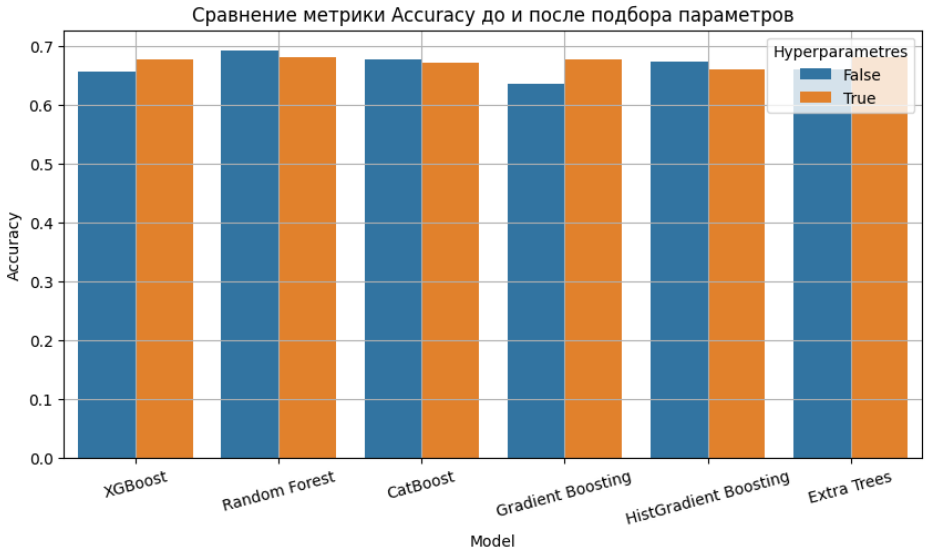

Рисунок 4.3.1. Сравнение метрики Accuracy до и после подбора гиперпараметров

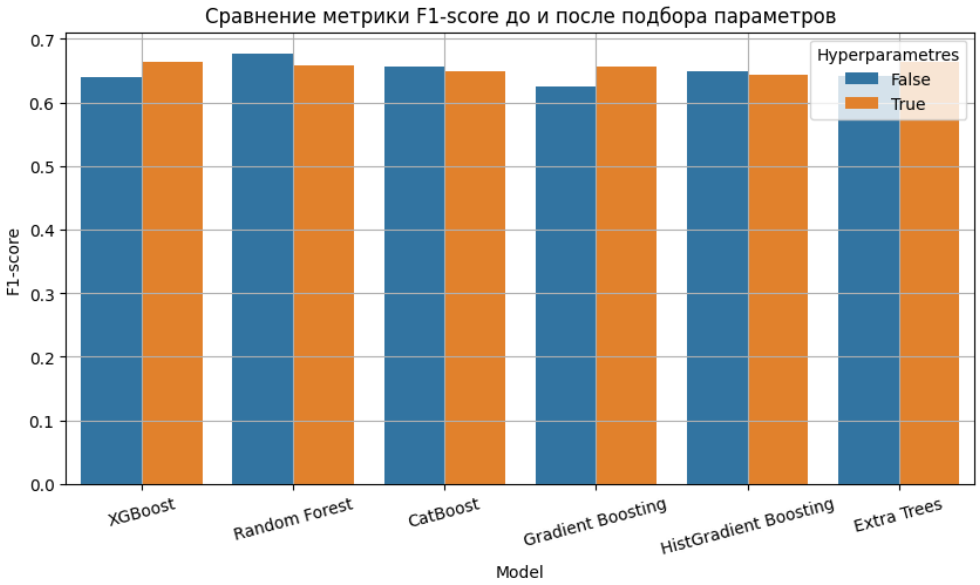

Рисунок 4.3.2. Сравнение метрики F1-score до и после подбора гиперпараметров

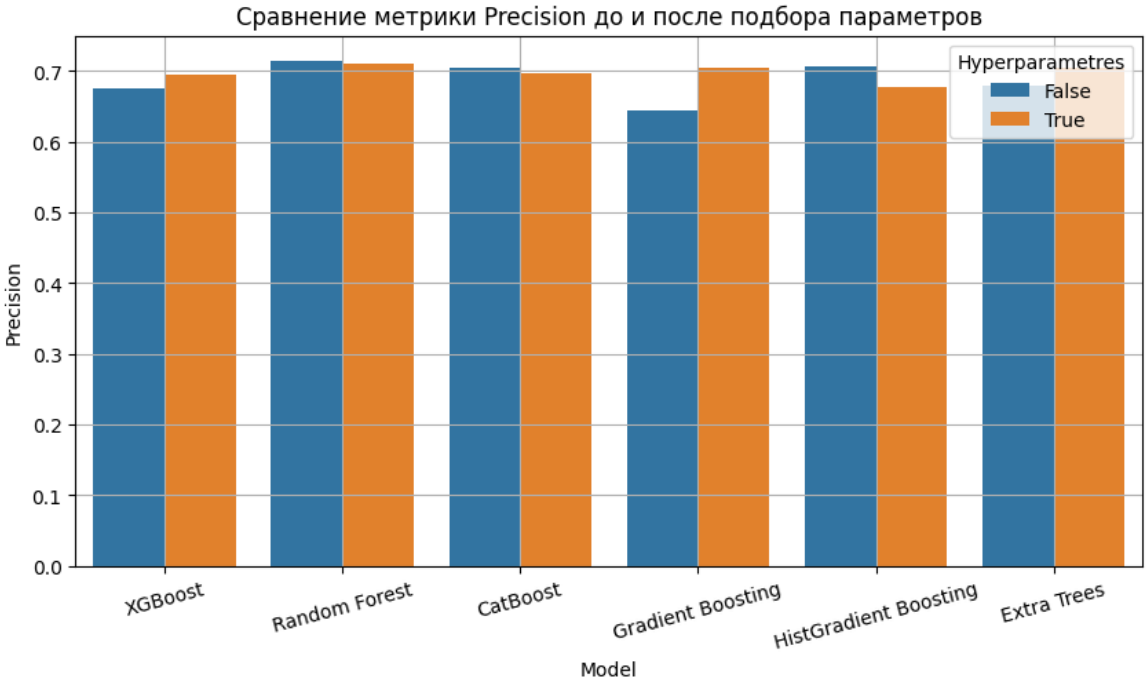

Рисунок 4.3.3. Сравнение метрики Precision до и после подбора гиперпараметров

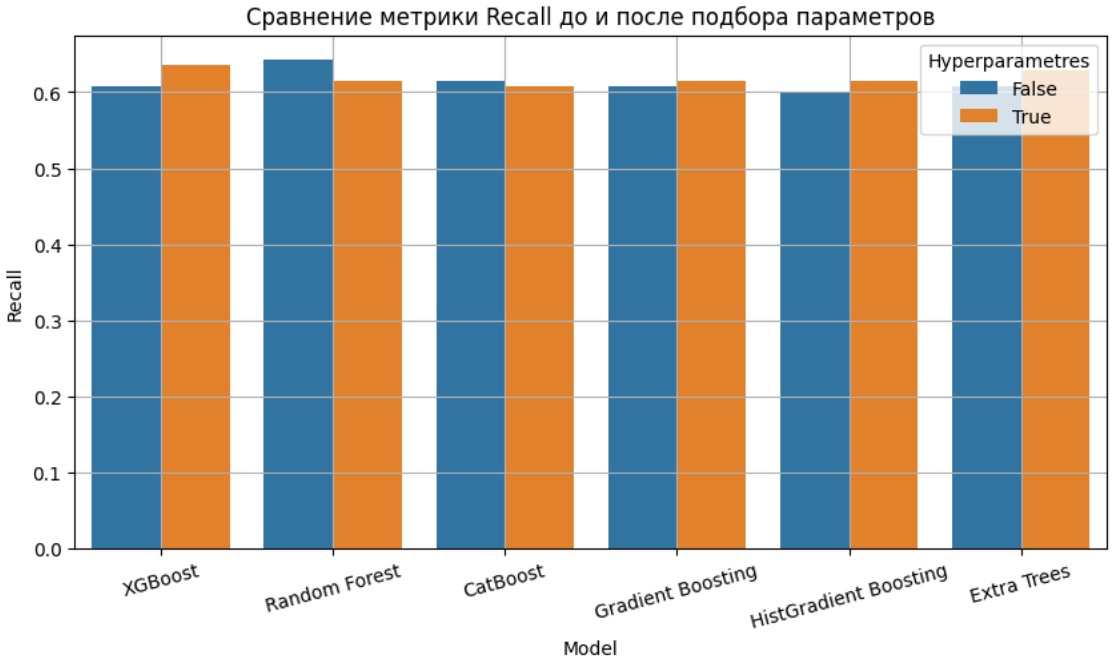

Рисунок 4.3.4. Сравнение метрики Recall до и после подбора гиперпараметров

Сводка по метрикам

Метрика: Accuracy

- Лучшая модель: Random Forest → 0.6918
- Худшая модель: Gradient Boosting → 0.6344
- Модели без улучшений: CatBoost, HistGradient Boosting, Random Forest

Метрика: F1-score

- Лучшая модель: Random Forest → 0.6767
- Худшая модель: Gradient Boosting → 0.6250
- Без улучшений: CatBoost, HistGradient Boosting, Random Forest

Метрика: Precision

- Лучшая модель: Random Forest → 0.7143

- Худшая модель: Gradient Boosting → 0.6439
- Без улучшений: CatBoost, HistGradient Boosting, Random Forest

Метрика: Recall

- Лучшая модель: Random Forest → 0.6429
- Худшая модель: HistGradient Boosting → 0.6000
- Без улучшений: CatBoost, Random Forest

Финальная настройка с Optuna

Оптимизация гиперпараметров с помощью Optuna позволила улучшить модель Random Forest, повысив Accuracy до 0.6989 и F1-score до 0.6794, что подтверждает эффективность подхода. Это подтверждает эффективность выбранного подхода и выделяет Random Forest как предпочтительный вариант для решения задачи классификации превышения медианного значения SI.

Финальные метрики после Optuna:

Accuracy	F1-score	Precision	Recall
0.6989	0.6794	0.7295	0.6357

Визуализация демонстрирует постепенное улучшение качества модели Random Forest на каждом этапе — от дефолтных параметров до финальной настройки с Optuna.

Рисунок 4.2.5. Изменение метрики Accuracy по этапам

Рисунок 4.2.6. Изменение метрики F1-score по этапам

Рисунок 4.2.7. Изменение метрики Precision по этапам

Рисунок 4.2.8. Изменение метрики Recall по этапам

Вывод по модели классификации

Исходя из анализа, оптимальной моделью для классификации превышения медианного значения SI является Random Forest с настройкой гиперпараметров через Optuna. Модель демонстрирует высокую точность и сбалансированность между полнотой и точностью, что важно для надёжного предсказания.

4.4 Создание модели классификации «Превышение SI 8»

Решение проблемы с дисбалансом классов

При анализе баланса классов, оказалось, что присутствует существенный дисбаланс (65 на 35), вследствие этого было принято решение провести корректировку.

Поскольку объем выборки относительно небольшой, метод undersampling был отклонён, так как это привело бы к потере значительной части данных. Вместо этого применялся oversampling с помощью метода SMOTE, который позволяет синтетически увеличивать меньший класс, не создавая копий, а генерируя новые наблюдения по близким точкам.

Результаты до подбора гиперпараметров

Модели обучались с использованием базовых параметров. Наилучший результат Accuracy показали модели XGBoost и CatBoost — 0.7025. Остальные модели показали схожие, но немного более низкие результаты.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
XGBoost	0.7025	0.5202	0.6000	0.4592	False
CatBoost	0.7025	0.5146	0.6027	0.4490	False
Random Forest	0.6882	0.5085	0.5696	0.4592	False
Gradient Boosting	0.6846	0.5111	0.5610	0.4694	False
Extra Trees	0.6846	0.5056	0.5625	0.4592	False
HistGradient Boosting	0.6810	0.5189	0.5517	0.4898	False

Результаты после RandomizedSearch

После настройки гиперпараметров произошло небольшое улучшение. CatBoost сохранил лучшие метрики F1-score и Recall, в то время как XGBoost продемонстрировал ухудшение по всем показателям. Некоторые модели, такие как Extra Trees и HistGradient Boosting, показали улучшение Precision, но не F1.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
CatBoost	0.6774	0.5213	0.5444	0.5000	True
HistGradient Boosting	0.6989	0.5116	0.5946	0.4490	True
Extra Trees	0.6810	0.5083	0.5542	0.4694	True
Gradient Boosting	0.6810	0.4795	0.5616	0.4184	True

Random Forest	0.6738	0.4740	0.5467	0.4184	True
XGBoost	0.6487	0.4432	0.5000	0.3980	True

Общая сводка

Из сводной таблицы видно, что наилучшие результаты до настройки гиперпараметров показала модель XGBoost (Accuracy = 0.7025), однако после настройки её качество снизилось по всем метрикам, особенно по Recall. В то же время CatBoost, напротив, улучшил F1-score и Recall после подбора параметров, хотя Accuracy незначительно просело. Также заметно, что улучшения от настройки параметров были в целом умеренными для всех моделей.

Model	Accuracy	F1-score	Precision	Recall	Hyperparameters
XGBoost	0.7025	0.5202	0.6000	0.4592	False
CatBoost	0.7025	0.5146	0.6027	0.4490	False
HistGradient Boosting	0.6989	0.5116	0.5946	0.4490	True
Random Forest	0.6882	0.5085	0.5696	0.4592	False
Gradient Boosting	0.6846	0.5111	0.5610	0.4694	False
Extra Trees	0.6846	0.5056	0.5625	0.4592	False
HistGradient Boosting	0.6810	0.5189	0.5517	0.4898	False
Gradient Boosting	0.6810	0.4795	0.5616	0.4184	True
Extra Trees	0.6810	0.5083	0.5542	0.4694	True
CatBoost	0.6774	0.5213	0.5444	0.5000	True
Random Forest	0.6738	0.4740	0.5467	0.4184	True
XGBoost	0.6487	0.4432	0.5000	0.3980	True

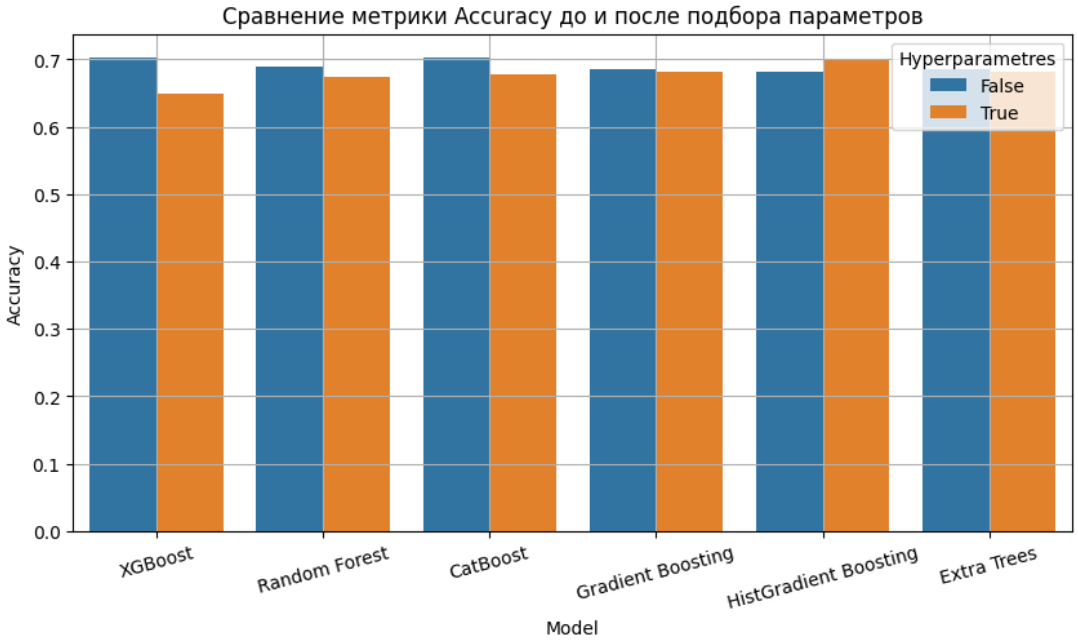

Рисунок 4.4.1. Сравнение метрики Accuracy до и после подбора гиперпараметров

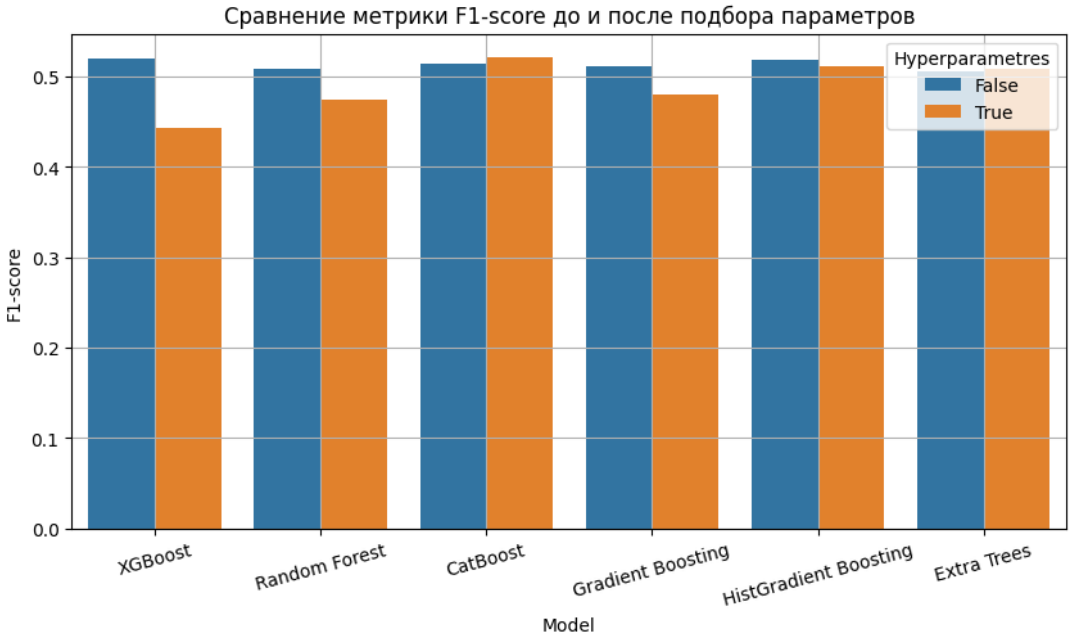

Рисунок 4.4.2. Сравнение метрики F1-score до и после подбора гиперпараметров

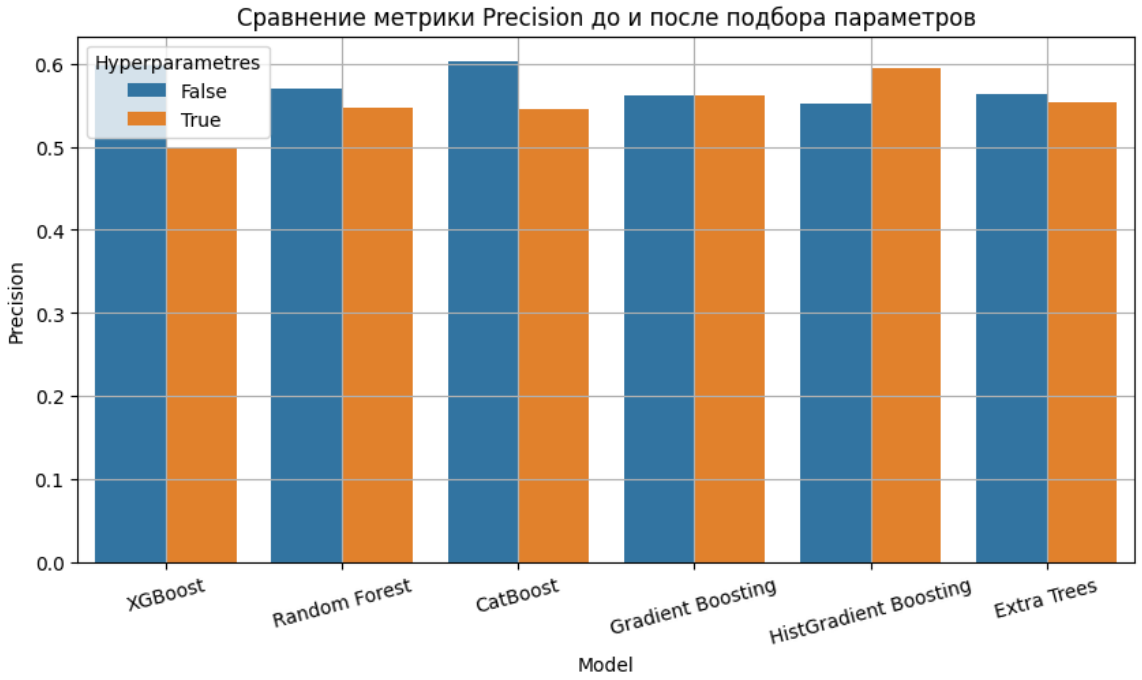

Рисунок 4.4.3. Сравнение метрики Precision до и после подбора гиперпараметров

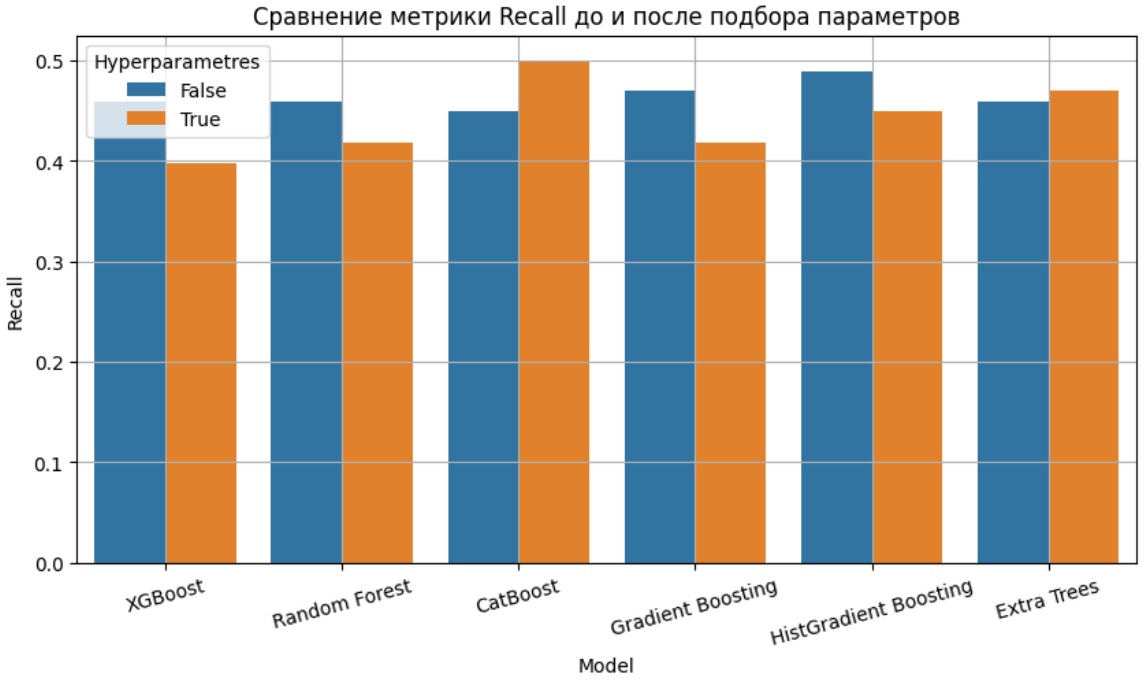

Рисунок 4.4.4. Сравнение метрики Recall до и после подбора гиперпараметров

Сводка по метрикам

Метрика: Accuracy

- Лучшая модель: XGBoost → 0.7025
- Худшая модель: XGBoost → 0.6487
- Без улучшений или хуже: CatBoost, Extra Trees, Gradient Boosting, Random Forest, XGBoost

Метрика: F1-score

- Лучшая модель: CatBoost → 0.5213
- Худшая модель: XGBoost → 0.4432
- Без улучшений или хуже: Gradient Boosting, HistGradient Boosting, Random Forest, XGBoost

Метрика: Precision

- Лучшая модель: CatBoost → 0.6027
- Худшая модель: XGBoost → 0.5000
- Без улучшений или хуже: CatBoost, Extra Trees, Random Forest, XGBoost

Метрика: Recall

- Лучшая модель: CatBoost → 0.5000
- Худшая модель: XGBoost → 0.3980
- Без улучшений или хуже: Gradient Boosting, HistGradient Boosting, Random Forest, XGBoost

Финальная настройка с Optuna

Оптимизация гиперпараметров с использованием Optuna позволила достичь умеренного улучшения по метрике Accuracy (до 0.7097). Однако метрика F1-score практически не изменилась (увеличение на 0.0005).

Финальные метрики после Optuna:

Accuracy	F1-score	Precision	Recall
0.7097	0.5207	0.6197	0.4490

Визуализация демонстрирует постепенное улучшение качества модели XGBoost на каждом этапе — от дефолтных параметров до финальной настройки с Optuna.

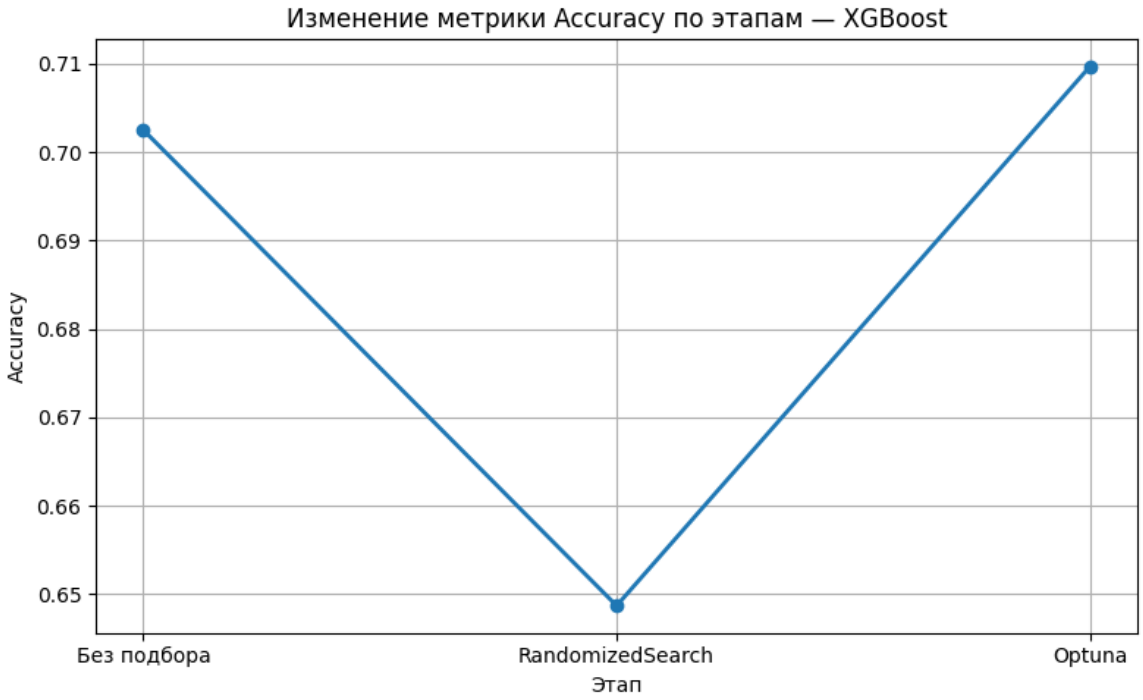

Рисунок 4.2.5. Изменение метрики Accuracy по этапам

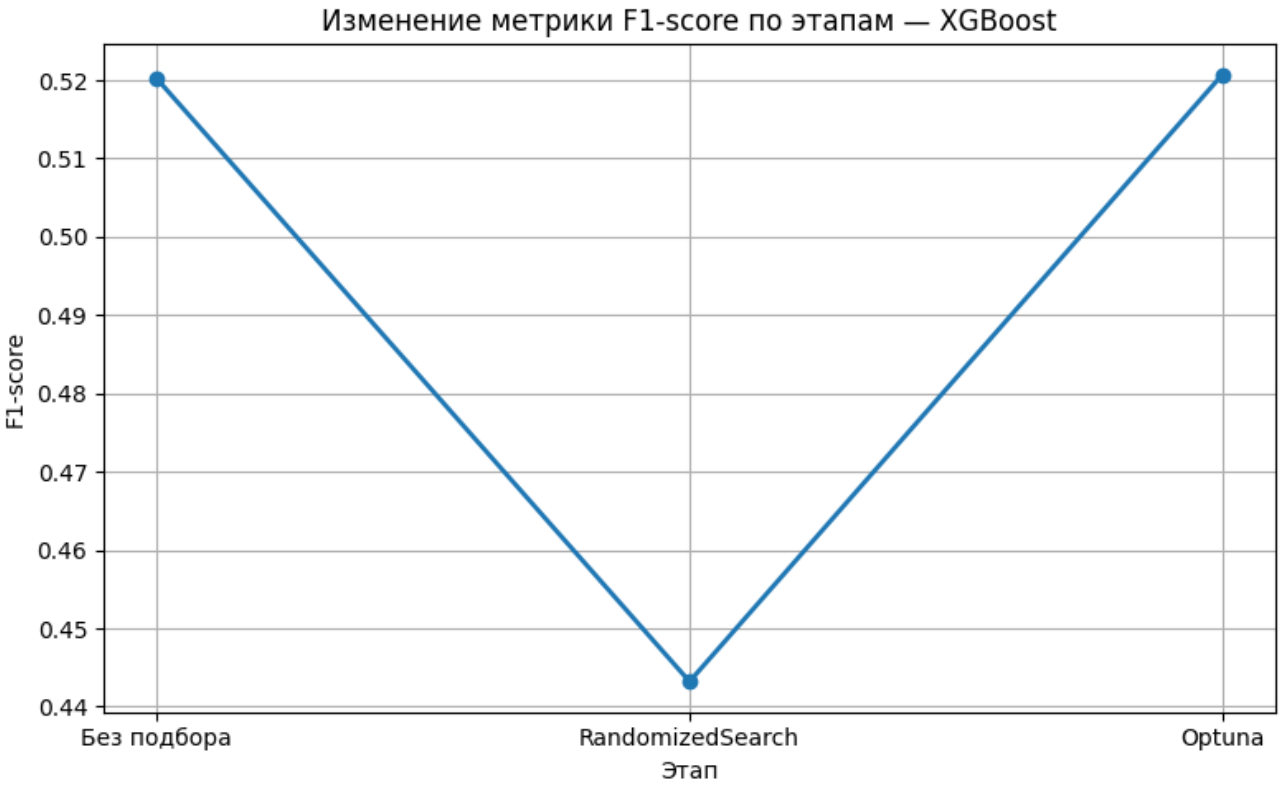

Рисунок 4.2.6. Изменение метрики F1-score по этапам

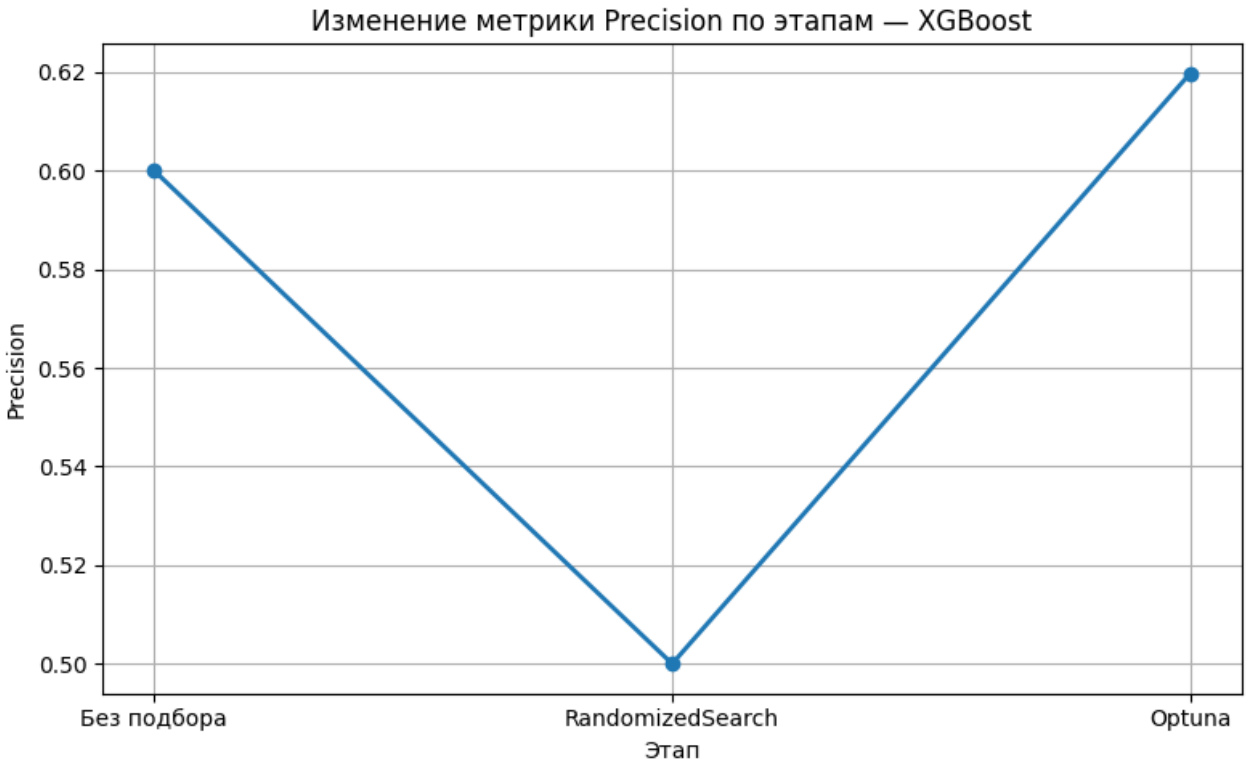

Рисунок 4.2.7. Изменение метрики Precision по этапам

Рисунок 4.2.8. Изменение метрики Recall по этапам

Вывод по модели классификации

Исходя из анализа, оптимальной моделью для классификации превышения значения в столбце SI числа 8 является XGBoost с настройкой гиперпараметров через Optuna. Модель демонстрирует высокую точность и сбалансированность между полнотой и точностью, что важно для надёжного предсказания.

4. Выводы

В рамках курсовой работы была проведена комплексная разработка и сравнение моделей машинного обучения для задач регрессии и классификации, направленных на оценку биологической активности соединений по параметрам IC50, CC50 и SI. Всего было решено семь задач — три задачи регрессии и четыре задачи бинарной классификации.

На основе тщательного анализа метрик качества и результатов гиперпараметрической оптимизации были выбраны финальные модели:

- **Задачи регрессии:**
  - IC50: наилучшие результаты показала модель XGBoost, демонстрируя высокую точность предсказаний.
  - CC50: лучшей моделью стала Extra Trees, обеспечившая баланс между скоростью обучения и точностью.
  - SI: предпочтение отдано Gradient Boosting, благодаря стабильным результатам и хорошему приближению значений.
- **Задачи классификации:**
  - Превышение медианного значения IC50: лидирующую позицию заняла модель XGBoost, эффективно разделяя классы.
  - Превышение медианного значения CC50: лучшей оказалась XGBoost, особенно после настройки с помощью Optuna.
  - Превышение медианного значения SI: выбрана Random Forest.
  - Превышение SI 8: наилучшие результаты продемонстрировала XGBoost, став наиболее устойчивой и надёжной моделью в условиях дисбаланса классов.

Таким образом, XGBoost зарекомендовала себя как наиболее универсальная и эффективная модель в рамках данной задачи, показав высокие результаты как в задачах регрессии, так и в классификации. Проведённая работа подчёркивает важность выбора подходящих моделей и методов обработки данных (включая SMOTE и гиперпараметрическую оптимизацию) для получения надёжных и интерпретируемых результатов при анализе биологической активности химических соединений.