1. Постановка задачи

В данном проекте перед нами стоит цель разработать модели, позволяющие прогнозировать эффективность химических соединений против вируса гриппа и выбирать наиболее перспективные соединения для дальнейших лабораторных исследований и разработки лекарственных препаратов. Для этого нам предоставлены данные о 1000 химических соединениях. Для каждого из них в исходном наборе данных известны значения IC50, CC50 и SI. На их основе требуется:

1. Проанализировать распределения и взаимосвязи между признаками (включая дополнительные физико-химические свойства соединений, если они представлены), выявить возможные выбросы, аномалии и зависимости, которые могут помочь в дальнейшем построении более точных моделей.
2. Построить и сравнить несколько моделей регрессии для прогнозирования непрерывных показателей:
   - Регрессия для IC50;
   - Регрессия для CC50;
   - Регрессия для SI.

   В каждой задаче подобрать и протестировать разные алгоритмы, настроить гиперпараметры и оценить качество модели с помощью метрик (RMSE, MAE, R2).
3. Построить и сравнить несколько моделей классификации на основе бинарных целевых меток, сформированных по следующим правилам:
   - «IC50 > медиана выборки?»;
   - «CC50 > медиана выборки?»;
   - «SI > медиана выборки?»;
   - «SI > 8?».

   В каждой задаче подобрать и протестировать разные алгоритмы, настроить гиперпараметры и оценить качество модели с помощью метрик (f1 score, accuracy score, roc auc score).
4. Сравнить между собой полученные модели по их метрикам качества. На основании этого сделать обоснованный выбор лучших моделей для каждой из поставленных задач.

2. Описание и обработка данных (EDA)

2.1 Структура исходного набора данных

Предоставленные данные содержат информацию о 1000 химических соединениях, представленных в виде числовых дескрипторов, отражающих молекулярную структуру и физико-химические свойства. Целевые переменные включают:

- **IC50** — показатель ингибирующей активности;
- **CC50** — показатель цитотоксичности;
- **SI** — индекс селективности, рассчитываемый как отношение CC50 к IC50.

Признаки охватывают физико-химические, электронные, топологические и структурные характеристики, включая общие свойства, зарядовое распределение, структурные индексы, отпечатки и фрагментные дескрипторы.

Таким образом, предварительно датасет состоял из 214 столбцов (целевых переменных + числовых дескрипторов), без строковых признаков.

2.2 Работа с пропущенными значениями

При первичном просмотре датафрейма пропущенные значения обнаружились в следующих колонках:

'MaxPartialCharge', 'MinPartialCharge', 'MaxAbsPartialCharge', 'MinAbsPartialCharge', 'BCUT2D_MWHI', 'BCUT2D_MWLOW', 'BCUT2D_CHGHI', 'BCUT2D_CHGLO', 'BCUT2D_LOGPHI', 'BCUT2D_LOGPLOW', 'BCUT2D_MRHI', 'BCUT2D_MRLOW'

Для заполнения пропусков применялся алгоритм IterativeImputer, такой метод позволил прогнозировать недостающие численные значения с учётом взаимозависимостей между дескрипторами, а не просто заменять их средним или медианой.

2.3 Удаление нерелевантных и «плохих» признаков

В ходе анализа с использованием автоматического набора тестов из библиотеки deepchecks были выявлены следующие проблемы и рекомендации:

1. **Столбцы с единственным значением**
   Обнаружено 18 колонок, в которых все строки имели идентичное значение. Такие признаки не несут никакой дисперсионной информации, поэтому было решено их удалить.

Перечень удалённых 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'.

2. Обнаружено много пар столбцов, имеющих высокий уровень корреляции, для их удаления был использован класс DropCorrelatedFeatures с параметром threshold=0.9. Такой шаг позволяет избежать мультиколлинеарности.
3. В каждой колонке было не более одного типа пропусков — что соответствует требованиям корректной табличной структуры;
4. Смешения типов данных (например, строки в числовой колонке) первичным скриптом не выявлено: все 214 столбцов (до удаления) были либо «чисто числовыми», либо содержали допустимые вариации;
5. Дубликаты строк не обнаружены, что гарантирует отсутствие дублирования образцов;
6. Строковых столбцов, где можно было бы проверить «длину строки», в данных не было, поэтому дополнительных выбросов по «длине строк» не выявлено.

Также был удален столбец 'Unnamed: 0', который содержал лишь индексы и не нес никакой смысловой нагрузки.

Далее была выполнена проверка IC50, CC50, и SI, которые по своей природе должны быть строго положительными, Все строки, в которых хотя бы одно из этих трёх значений оказалось ≤ 0, были удалены из датасета.

В результате всех перечисленных манипуляций из исходных 214 столбцов осталось 156 признаков (включая три целевых: IC50, CC50, SI и 153 числовых дескриптора).

2.4 Анализ распределений

Для каждой из трёх целевых переменных (IC50, CC50, SI) была проведена визуализация распределения (рис. 2.4.1).

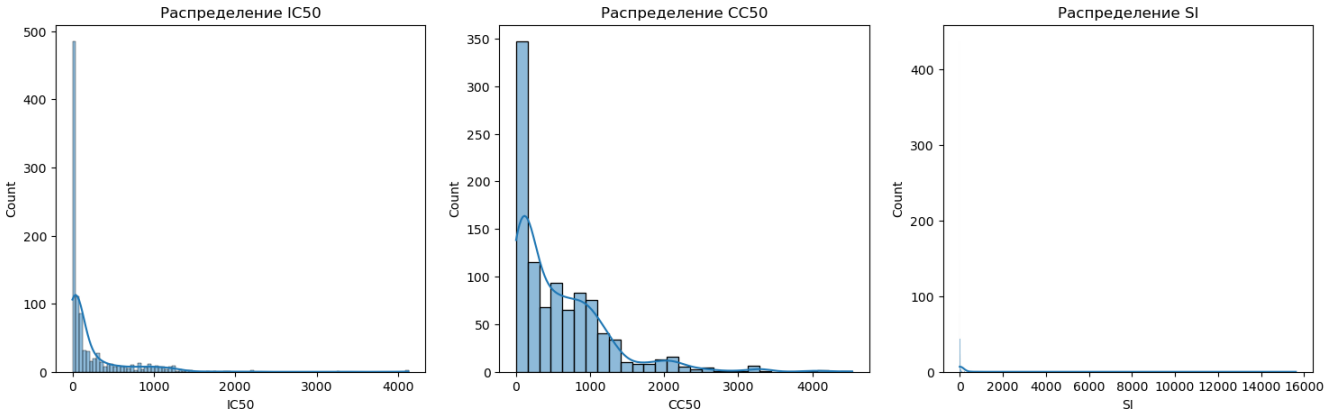

Рисунок 2.4.1 – Распределение целевых переменных

Из графиков мы можем сделать следующие выводы:

- Распределения имеют выраженную правостороннюю асимметрию
- Большая часть значений сосредоточена в нижних диапазонах, в то время как «длинный правый хвост» уходит к очень большим значениям (например, IC50 до 4000).

Для оценки распределения переменных IC50, CC50 и SI был проведён автоматический подбор наиболее подходящих статистических распределений с помощью библиотеки fitter. Результаты показали, что ни одна из переменных не подчиняется нормальному распределению (рис. 2.4.2).

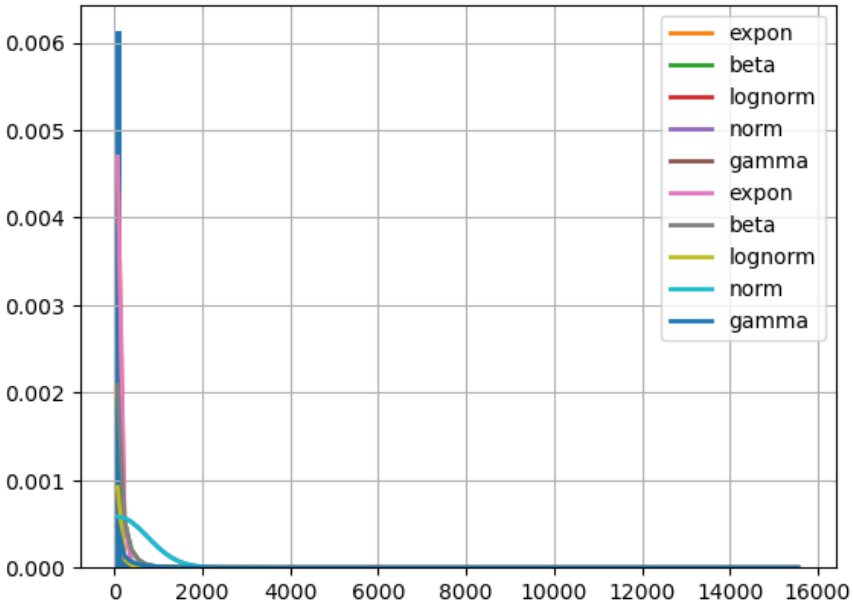

Рисунок 2.4.2 – Сравнение распределений для целевых переменных

Можно сделать следующие выводы:

- IC50 наиболее близко к lognorm распределению, что подтверждается минимальной ошибкой подгонки среди рассматриваемых распределений.
- CC50 показывает хорошее соответствие с гамма и бета-распределениями, а также логнормальному, что свидетельствует о положительной асимметрии данных.
- SI наиболее близко к бета-распределению.

2.5. Анализ выбросов

Для каждой из трёх переменных были построены boxplot-диаграммы (рис. 2.4.3), позволяющие наглядно увидеть наличие точек-выбросов за пределами усов.

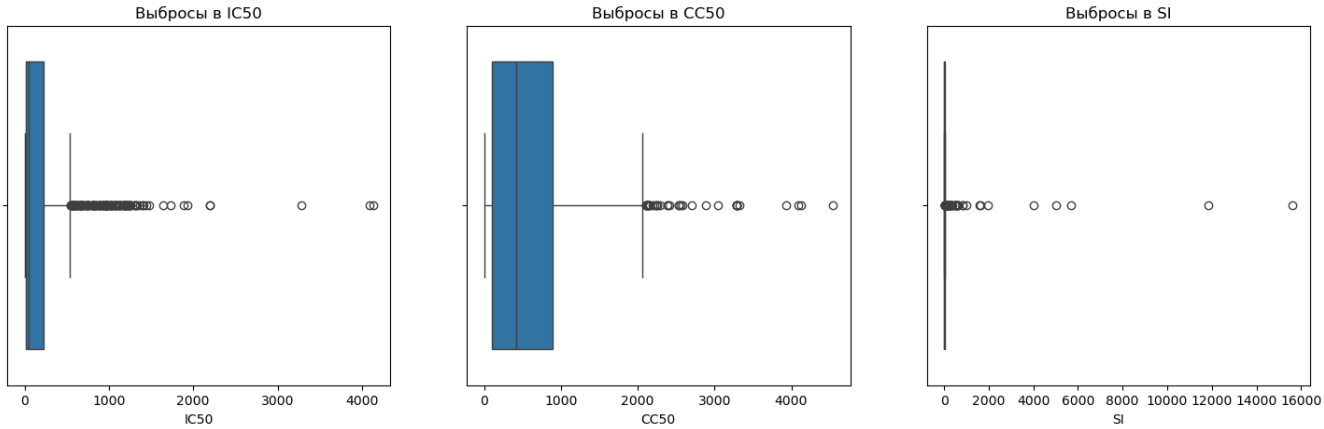

Рисунок 2.4.3 – Boxplot-диаграммы IC50, CC50 и SI

Из рисунка видно:

- Во всех трёх случаях выбросы проявляются исключительно справа. Левый ус не имеет точек-выбросов, так как минимальные наблюдаемые значения не уходят далеко за нижний квартиль.
- Межквартильный интервал указывает, что большая часть данных находится в нижних и средних диапазонах, а экстремальные значения формируют правосторонние выбросы.

Таким образом, визуализация подтверждает: выбросы следует удалять исключительно по правому хвосту, поскольку слева (на малых значениях) экстремальных отклонений нет.

Для снижения влияния экстремальных значений, применён классический метод удаления выбросов по межквартильному размаху (IQR). Алгоритм удаления состоит из следующих шагов:

1. Для выбранной целевой переменной (например, IC50) вычисляются первый и третий квартиль.
2. Рассчитывается межквартильный размах:

$$IQR = Q_3 - Q_1$$

3. Определяется верхняя граница выбросов:

$$upper\ bound = \mathrm{Q3} + 1.5 \times \mathrm{IQR}$$

4. Все образцы, у которых значение целевой метрики превышает верхнюю границу, считаются выбросами и отфильтровываются.

Этот метод сохранит основную часть данных и удалит только самые далекие «хвосты».
После фильтрации выбросов были построены повторные boxplot-диаграммы (рис. 2.5.1), демонстрирующие, как изменились распределения целевых переменных без экстремальных правосторонних значений.

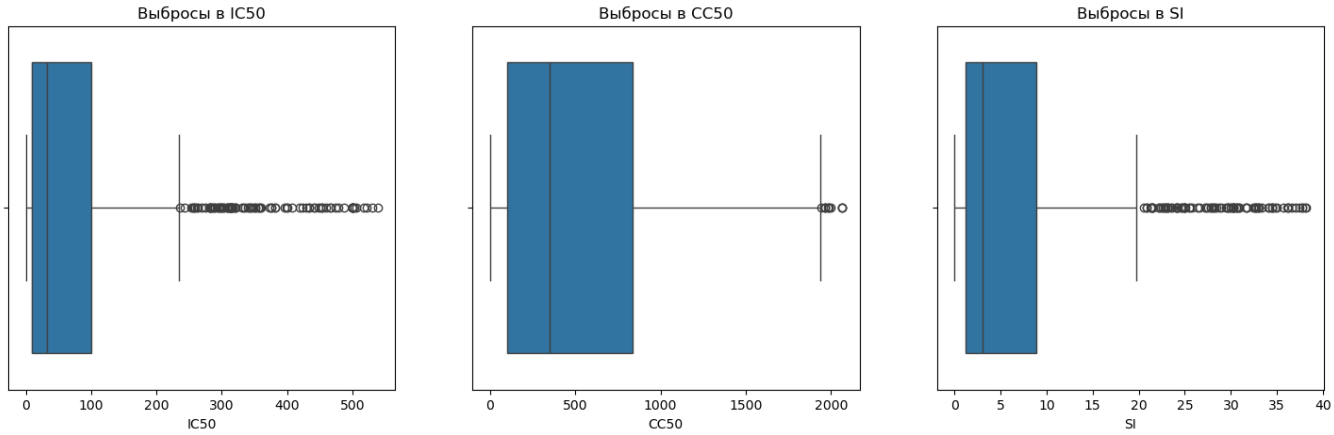

Рисунок 2.4.4 – Boxplot-диаграммы IC50, CC50 и SI после удаления выбросов

Из нового набора диаграмм видно:

- Правая граница «уса» значительно сократилась, выбросы, выходящие далеко вправо, отсутствуют.
- Левая сторона распределений по-прежнему не содержит выбросов, что согласуется с предыдущим анализом.

После применения IQR-метода получены три очищенных датафрейма, и ниже приведены полученные результаты:

- **IC50**
  - Было строк: 1000
  - Строк осталось после фильтрации: 854 (85.4%)
  - Количество удалённых выбросов: 146 (14.6%)
- **CC50**
  - Было строк: 1000
  - Строк осталось после фильтрации: 962 (96.2%)
  - Количество удалённых выбросов: 38 (3.8%)
- **SI**
  - Было строк: 1000
  - Строк осталось после фильтрации: 876 (87.6%)
  - Количество удалённых выбросов: 124 (12.4%)

3. Построение моделей регрессии

Целевые переменные обладают ненормальным распределением с выраженной правосторонней асимметрией и длинным правым хвостом. Такие распределения осложняют применение моделей, требующих нормальности ошибок (например, линейной регрессии). В связи с этим при построении моделей основное внимание уделяется алгоритмам на основе деревьев решений (Random Forest, градиентный бустинг и др.), которые менее чувствительны к форме распределения данных.

Так как все перечисленные алгоритмы основаны на деревьях решений, они устойчивы к масштабу признаков и не требуют предварительной стандартизации или нормализации данных. Кроме того, методы снижения размерности, такие как PCA, не являются обязательными, поскольку деревья эффективно работают с избыточностью и коррелированными признаками.

Поэтому подготовленные данные подавались в модели напрямую, без предварительной стандартизации или понижения размерности.

Для сравнения рассматриваются следующие модели:

- XGBRegressor,
- XGBRegressor с функцией потерь Tweedie,
- RandomForestRegressor,
- GradientBoostingRegressor,
- CatBoostRegressor,
- CatBoostRegressor с функцией потерь Tweedie
- HistGradientBoostingRegressor,
- ExtraTreesRegressor.

Пайплайн обработки и оценки моделей будет одинаков для всех трёх целевых переменных:

Данные сначала делятся на обучающую и тестовую выборки с помощью метода train-test split. На начальном этапе модели обучаются на тренировочном поднаборе, и по результатам их работы вычисляются базовые метрики качества.

Для повышения качества моделей проводится оптимизация гиперпараметров с помощью библиотеки Optuna.

Для оптимизации гиперпараметров с помощью Optuna я выбрал метрику R2 по следующим причинам:

1. Интерпретируемость и смысл — R2 показывает долю объяснённой дисперсии, то есть насколько модель улучшает предсказания по сравнению с простым усреднением. Это даёт более наглядную оценку качества модели, чем абсолютные ошибки.
2. Оптимизация качества объяснения — при подборе гиперпараметров важна максимизация способности модели объяснять вариации данных, а не только минимизация ошибки. R2 именно это отражает.

Для каждой модели создаётся отдельный процесс настройки, включающий:

1. Повторное разделение тренировочного набора на поднаборы для обучения и валидации (80% и 20% соответственно).
2. Многошаговый перебор гиперпараметров, где на каждом шаге модель обучается на тренировочной части и оценивается по метрике R2 на валидационной.
3. Выбор оптимального набора гиперпараметров, обеспечивающего максимальное значение R2 на валидационной выборке.
4. Обучение финальной модели с подобранными параметрами на полном тренировочном наборе.
5. Оценка итоговой модели на тестовых данных по метрикам R2, RMSE и MAE.

Вся процедура реализована в функции optuna_tuning, которая принимает словарь моделей, последовательно оптимизирует каждую из них и возвращает итоговую таблицу с метриками и обученные модели.

Функция optuna_tuning:

- Для каждой модели выполняет повторное разбиение данных на обучающую и валидационную части.
- Определяет функцию цели для Optuna, которая настраивает специфичные для модели гиперпараметры.
- Запускает оптимизацию для поиска наилучших параметров по метрике R2.
- После оптимизации обучает финальную модель на полном тренировочном наборе.
- Оценивает модель на тестовом наборе и сохраняет метрики и обученный объект.

Таким образом, подход обеспечивает тщательный подбор гиперпараметров и справедливую оценку качества моделей на отложенных данных.

3.1 Регрессия для IC50

Сначала проведем тестирование моделей с базовыми настройками, результаты метрик без оптимизации гиперпараметров представлены в таблице 3.1.1.

Таблица 3.1.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	R2	RMSE	MAE
XGB	0.1210	97.1841	58.8789
XGB_Tweedie	0.3023	86.5880	48.1642
RandomForest	0.1687	94.5122	58.7291
GradientBoosting	0.1889	93.3546	61.9057
CatBoost	0.2102	92.1207	57.8841
CatBoost_Tweedie	0.3215	85.3831	48.0945
HistGradientBoostingRegressor	0.0070	103.2943	64.3260
ExtraTreesRegressor	0.2115	92.0490	58.9141

- Лучшие результаты показали модели CatBoost_Tweedie и XGB_Tweedie — у них наибольшие значения R2 (0.3215 и 0.3023 соответственно), а также наименьшие значения RMSE (85.38 и 86.59) и MAE (48.09 и 48.16).
- Худшие результаты у модели HistGradientBoostingRegressor: минимальное R2 (0.0070) и наибольшие ошибки (RMSE — 103.29, MAE — 64.33), что говорит о слабой пригодности этой модели без настройки гиперпараметров.

Визуальное сравнение по метрикам представлено на рисунке 3.1.1.

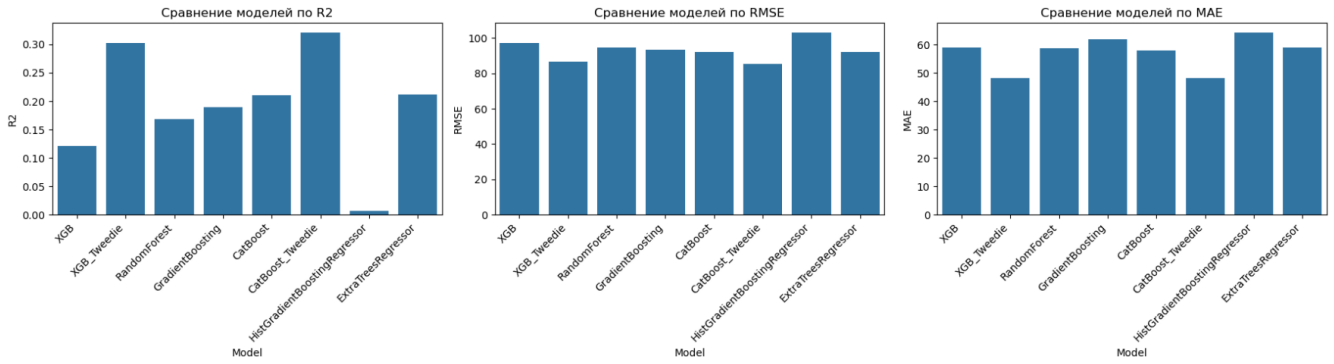

Рисунок 3.1.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 3.1.2.

Таблица 3.1.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	R2	RMSE	MAE
XGB	0.1965	92.9168	59.4386
XGB_Tweedie	0.2716	88.4664	55.0039
RandomForest	0.2071	92.3055	59.0738
GradientBoosting	0.2428	90.2010	59.2880
CatBoost	0.2889	87.4114	57.2577
CatBoost_Tweedie	0.3179	85.6113	53.3481
HistGradientBoostingRegressor	0.1739	94.2133	62.8600
ExtraTreesRegressor	0.2758	88.2144	59.4802

- Наилучшие результаты снова показала модель CatBoost_Tweedie — у неё самое высокое значение R2 (0.3179), а также наименьшие RMSE (85.61) и MAE (53.35). Также хорошие показатели у CatBoost и ExtraTreesRegressor.
- Худшие результаты по-прежнему у HistGradientBoostingRegressor — самый низкий R2 (0.1739), высокий RMSE (94.21) и MAE (62.86), несмотря на настройку параметров.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 3.1.2.

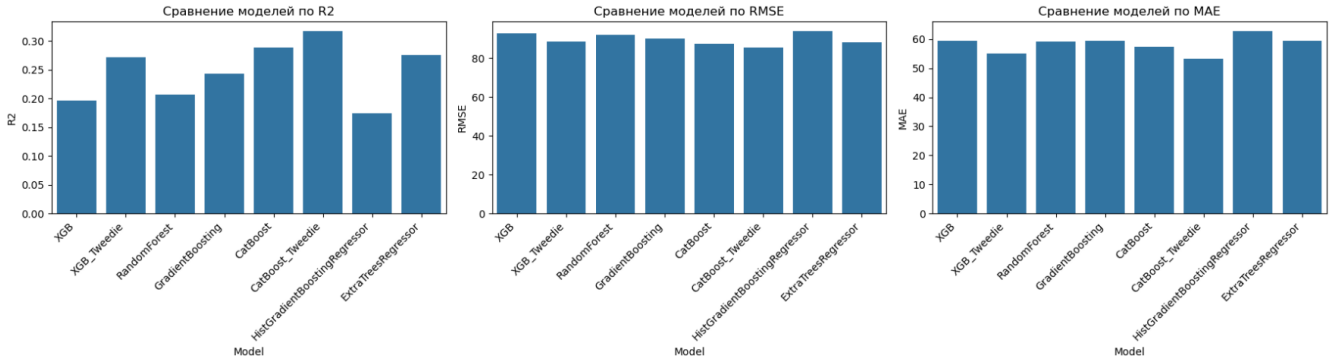

Рисунок 3.1.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 3.1.3):

Таблица 3.1.3 – Общие результаты моделей.

Модель	R2	RMSE	MAE	Tuned
CatBoost_Tweedie	0.3215	85.3831	48.0945	False
CatBoost_Tweedie	0.3179	85.6113	53.3481	True
XGB_Tweedie	0.3023	86.5880	48.1642	False

Модель	R2	RMSE	MAE	Tuned
CatBoost	0.2889	87.4114	57.2577	True
ExtraTreesRegressor	0.2758	88.2144	59.4802	True
XGB_Tweedie	0.2716	88.4664	55.0039	True
GradientBoosting	0.2428	90.2010	59.2880	True
ExtraTreesRegressor	0.2115	92.0490	58.9141	False
CatBoost	0.2102	92.1207	57.8841	False
RandomForest	0.2071	92.3055	59.0738	True
XGB	0.1965	92.9168	59.4386	True
GradientBoosting	0.1889	93.3546	61.9057	False
HistGradientBoostingRegressor	0.1739	94.2133	62.8600	True
RandomForest	0.1687	94.5122	58.7291	False
XGB	0.1210	97.1841	58.8789	False
HistGradientBoostingRegressor	0.0070	103.2943	64.3260	False

- Лучшие результаты показала модель CatBoost_Tweedie без подбора гиперпараметров, продемонстрировав наивысшее значение R2 (0.3215) и наименьшие значения RMSE (85.38) и MAE (48.09). Схожие по качеству прогнозирования результаты также у её версии с подбором и у XGB_Tweedie без подбора.
- Худшие показатели у HistGradientBoostingRegressor без подбора — минимальное R2 (0.0070), а также максимальные значения ошибок RMSE (103.29) и MAE (64.33), что делает её наименее подходящей моделью в текущем сравнении.

В целом, Tweedie-модификации градиентного бустинга (особенно CatBoost_Tweedie) стабильно демонстрируют высокое качество предсказания как до, так и после настройки параметров. Оптимальной моделью для применения в задаче прогнозирования IC50 является CatBoost_Tweedie, обеспечивающая наилучший баланс между точностью и стабильностью результатов.

На рисунке 3.1.3 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперапарметров.

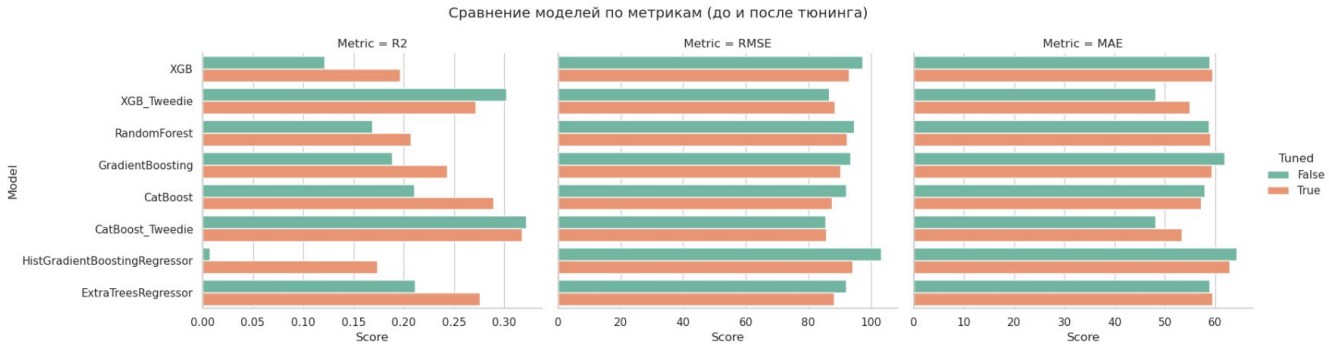

Рисунок 3.1.3– Сравнение моделей по метрикам до и после подбора гиперпараметров

3.2 Регрессия для CC50

Сначала проведем тестирование моделей с базовыми настройками, результаты метрик без оптимизации гиперпараметров представлены в таблице 3.2.1.

Таблица 3.2.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	R2	RMSE	MAE
XGB	0.5891	307.4675	189.0528
XGB_Tweedie	0.4637	351.2326	200.5619
RandomForest	0.6168	296.9035	196.7001
GradientBoosting	0.6246	293.8594	204.8962
CatBoost	0.6223	294.7615	193.1516
CatBoost_Tweedie	0.5112	335.3213	198.4659
HistGradientBoostingRegressor	0.5974	304.3362	197.2478
ExtraTreesRegressor	0.5788	311.2923	189.9262

- Лучшие результаты показала модель GradientBoosting с максимальным R2 (0.6246) и минимальным RMSE (293.86). Очень близкие значения у CatBoost и RandomForest, что также указывает на их высокую эффективность.
- Наименьшие ошибки MAE — у модели XGB (189.05) и ExtraTreesRegressor (189.93), при этом их R2 также остаются достаточно высокими.
- Худшие результаты по R2 у модели XGB_Tweedie (0.4637) и CatBoost_Tweedie (0.5112), несмотря на хорошие результаты этих моделей в задаче IC50. Это говорит о менее удачном применении Tweedie-регрессии для CC50 в текущих условиях.

Визуальное сравнение по метрикам представлено на рисунке 3.2.1.

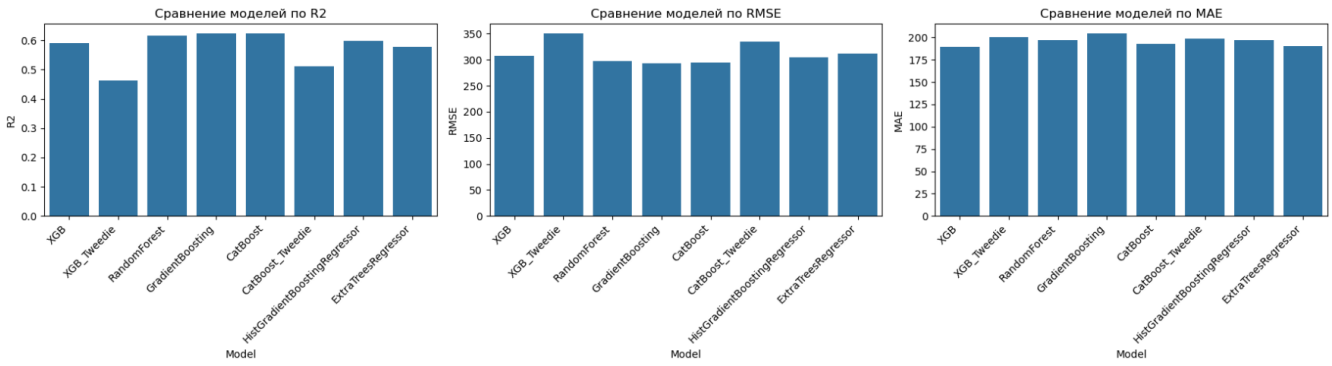

Рисунок 3.2.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 3.2.2.

Таблица 3.2.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	**R2**	**RMSE**	**MAE**
XGB	0.5950	305.2333	207.7947
XGB_Tweedie	0.5505	321.5631	202.4438
RandomForest	0.6107	299.2765	197.9957
GradientBoosting	0.5769	311.9868	214.9640
CatBoost	0.6396	287.9338	196.8751
CatBoost_Tweedie	0.5656	316.1061	197.3229
HistGradientBoostingRegressor	0.6111	299.0948	197.0752
ExtraTreesRegressor	0.6161	297.1661	189.8310

- Лучший результат показала модель CatBoost — наибольшее значение R2 (0.6396), а также минимальный RMSE (287.93), что указывает на высокую точность предсказаний.
- Минимальное значение MAE — у ExtraTreesRegressor (189.83), при этом её R2 (0.6161) также остаётся на высоком уровне, что делает модель хорошо сбалансированной.
- Наименее эффективной стала модель GradientBoosting — у неё наихудшее значение MAE (214.96) и сравнительно низкий R2 (0.5769) после настройки, что говорит о снижении её качества после подбора. Также XGB_Tweedie имеет наименьшее R2 (0.5505) среди всех моделей.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 3.2.2.

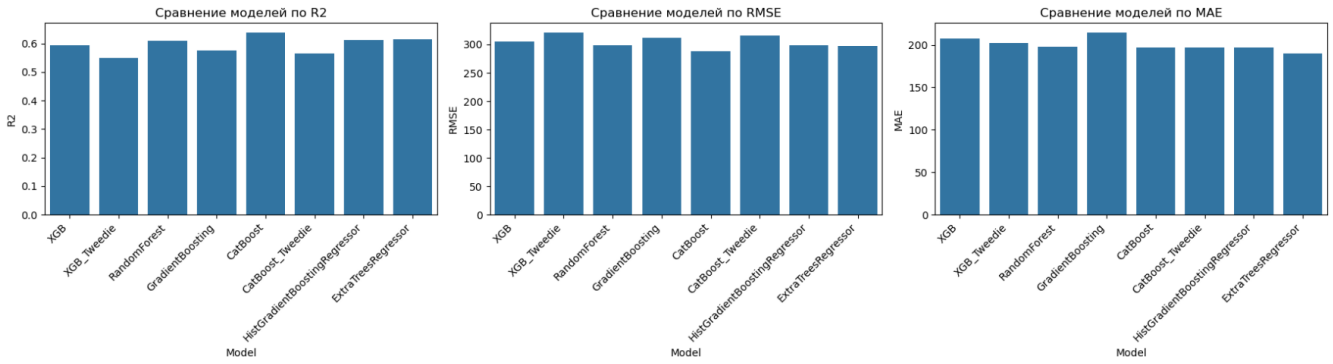

Рисунок 3.2.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 3.2.3):

Таблица 3.2.3 – Общие результаты моделей.

Модель	R2	RMSE	MAE	Tuned
CatBoost	0.6396	287.9338	196.8751	True
GradientBoosting	0.6246	293.8594	204.8962	False
CatBoost	0.6223	294.7615	193.1516	False
RandomForest	0.6168	296.9035	196.7001	False
ExtraTreesRegressor	0.6161	297.1661	189.8310	True
HistGradientBoostingRegressor	0.6111	299.0948	197.0752	True
RandomForest	0.6107	299.2765	197.9957	True
HistGradientBoostingRegressor	0.5974	304.3362	197.2478	False
XGB	0.5950	305.2333	207.7947	True
XGB	0.5891	307.4675	189.0528	False
ExtraTreesRegressor	0.5788	311.2923	189.9262	False
GradientBoosting	0.5769	311.9868	214.9640	True
CatBoost_Tweedie	0.5656	316.1061	197.3229	True
XGB_Tweedie	0.5505	321.5631	202.4438	True
CatBoost_Tweedie	0.5112	335.3213	198.4659	False
XGB_Tweedie	0.4637	351.2326	200.5619	False

- Лучшие результаты показала модель CatBoost с подбором гиперпараметров — максимальное R2 (0.6396), минимальный RMSE (287.93) и хорошие показатели MAE (196.88). Очень близкие результаты у GradientBoosting без

подбора и CatBoost без подбора, а также у RandomForest без подбора и ExtraTreesRegressor с подбором.

- Наихудшие показатели у моделей XGB_Tweedie без подбора и CatBoost_Tweedie без подбора, с наименьшими R2 (0.4637 и 0.5112) и высокими ошибками.

В целом, для задачи прогнозирования CC50 оптимальной моделью является CatBoost с подобранными гиперпараметрами, которая обеспечивает наилучший баланс точности и стабильности результатов.

На рисунке 3.2.3 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперпараметров.

Рисунок 3.2.3– Сравнение моделей по метрикам до и после подбора гиперпараметров

3.3 Регрессия для SI

Сначала проведем тестирование моделей с базовыми настройками, результаты метрик без оптимизации гиперпараметров представлены в таблице 3.3.1.

Таблица 3.3.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	R2	RMSE	MAE
XGB	-0.1277	9.0371	5.8764
XGB_Tweedie	-0.0228	8.6063	5.0845
RandomForest	0.0215	8.4181	5.8684
GradientBoosting	0.1142	8.0092	5.4552
CatBoost	0.0281	8.3895	5.5789
CatBoost_Tweedie	0.0735	8.1911	4.91
HistGradientBoosting	0.025	8.4031	5.5868
ExtraTreesRegressor	-0.0394	8.6761	5.6291

- Лучшие результаты у модели GradientBoosting — самое высокое значение R2 (0.1142) и одна из наименьших ошибок RMSE (8.01) и MAE (5.46). Также хорошие показатели у CatBoost_Tweedie — R2 (0.0735) с минимальным MAE (4.91).
- Худшие результаты у моделей XGB и ExtraTreesRegressor — отрицательные значения R2 (-0.1277 и -0.0394 соответственно), а также большие ошибки, что говорит о плохом качестве предсказаний без настройки гиперпараметров.

В целом, наблюдается слабая объяснённая дисперсия целевой переменной, что может указывать на сложность задачи и необходимость дальнейшего подбора гиперпараметров.

Визуальное сравнение по метрикам представлено на рисунке 3.3.1.

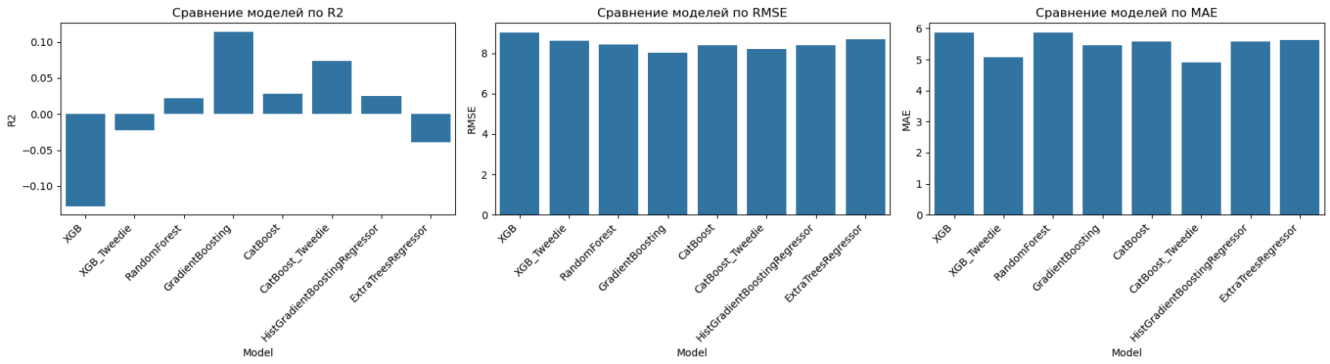

Рисунок 3.3.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 3.3.2.

Таблица 3.3.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	R2	RMSE	MAE
XGB	0.1407	7.8884	5.5434
XGB_Tweedie	0.091	8.1137	5.3519
RandomForest	0.1305	7.9352	5.715
GradientBoosting	0.1502	7.8448	5.4877
CatBoost	0.0833	8.1479	5.6424
CatBoost_Tweedie	0.1483	7.8538	5.1547
HistGradientBoostingRegressor	0.1349	7.9151	5.5755
ExtraTreesRegressor	0.1162	8.0003	5.6628

- Лучшие результаты показали модели GradientBoosting (наивысший R2 = 0.1502) и CatBoost_Tweedie (второй по R2 — 0.1483, но наименьший MAE — 5.15), обе с хорошими значениями RMSE (~7.84–7.85).
- Худшие результаты после подбора — у CatBoost (R2 = 0.0833, RMSE = 8.15, MAE = 5.64), несмотря на настройку гиперпараметров, он отстаёт от остальных моделей по всем метрикам.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 3.3.2.

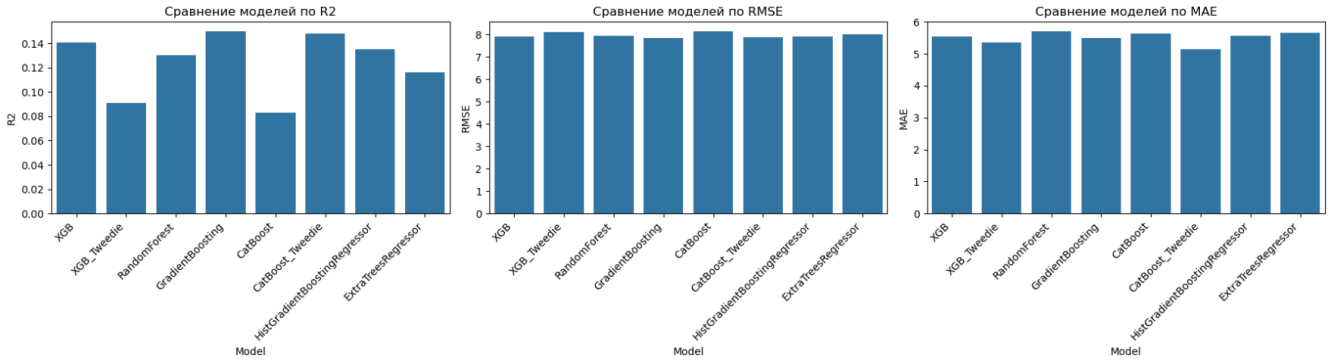

Рисунок 3.3.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 3.3.3):

Таблица 3.3.3 – Общие результаты моделей.

Модель	R2	RMSE	MAE	Tuned
GradientBoosting	0.1502	7.8448	5.4877	True
CatBoost_Tweedie	0.1483	7.8538	5.1547	True
XGB	0.1407	7.8884	5.5434	True
HistGradientBoostingRegressor	0.1349	7.9151	5.5755	True
RandomForest	0.1305	7.9352	5.715	True
ExtraTreesRegressor	0.1162	8.0003	5.6628	True
GradientBoosting	0.1142	8.0092	5.4552	False
XGB_Tweedie	0.091	8.1137	5.3519	True
CatBoost	0.0833	8.1479	5.6424	True
CatBoost_Tweedie	0.0735	8.1911	4.91	False
CatBoost	0.0281	8.3895	5.5789	False
HistGradientBoostingRegressor	0.025	8.4031	5.5868	False

Модель	R2	RMSE	MAE	Tuned
RandomForest	0.0215	8.4181	5.8684	False
XGB_Tweedie	-0.0228	8.6063	5.0845	False
ExtraTreesRegressor	-0.0394	8.6761	5.6291	False
XGB	-0.1277	9.0371	5.8764	False

- Лучшие результаты показали модели GradientBoosting (с подбором) и CatBoost_Tweedie (с подбором) — они обеспечили наивысшие значения R2 (0.1502 и 0.1483 соответственно), при этом CatBoost_Tweedie дал наименьший MAE (5.15), что особенно важно при оценке точности.
- Худшие показатели у XGB без подбора — отрицательное значение R2 (-0.1277), а также наибольшие ошибки RMSE (9.04) и MAE (5.88), что указывает на крайне низкое качество предсказаний этой модели без настройки.

В целом, подбор гиперпараметров значительно улучшает качество моделей в задаче прогнозирования SI. Среди всех алгоритмов оптимальной моделью можно считать CatBoost_Tweedie с подбором параметров, так как она сочетает высокую объясняющую способность и минимальные ошибки.

На рисунке 3.3.2 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперапарметров.

Рисунок 3.3.2– Сравнение моделей по метрикам до и после подбора гиперпараметров

4. Построение моделей классификации

Для сравнения рассматриваются следующие модели:

- XGBClassifier,
- RandomForestClassifier,
- GradientBoostingClassifier,
- CatBoostClassifier,
- HistGradientBoostingClassifier,
- ExtraTreesClassifier.

Процесс обработки данных и оценки моделей реализован аналогично этапам построения регрессионных моделей, с одним исключением: перед разделением данных на признаки (X) и целевую переменную (y) создается бинарная целевая переменная, после чего удаляется оригинальная переменная.

4.1 Классификация: превышает ли значение IC50 медианное значение выборки

Сначала проведем тестирование моделей с базовыми настройками, результаты метрик без оптимизации гиперпараметров представлены в таблице 4.1.1.

Таблица 4.1.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	Accuracy	ROC_AUC	F1
XGB	0.8012	0.8671	0.7875
RandomForest	0.7485	0.85	0.7226
GradientBoosting	0.7895	0.8618	0.7778
CatBoost	0.7836	0.8654	0.773
HistGradientBoosting	0.7953	0.8663	0.7799
ExtraTrees	0.7193	0.8229	0.6842

- Лучшие результаты показала модель XGB — наивысшая Accuracy (0.8012), ROC AUC (0.8671) и F1 (0.7875), что говорит о её высокой стабильности и точности.
- Хорошо себя показали также HistGradientBoosting и GradientBoosting, с чуть меньшими, но близкими значениями метрик.
- Худшие результаты у ExtraTrees — самая низкая Accuracy (0.7193), ROC AUC (0.8229) и F1 (0.6842), что указывает на её ограниченную эффективность без настройки.

Визуальное сравнение по метрикам представлено на рисунке 4.1.1.

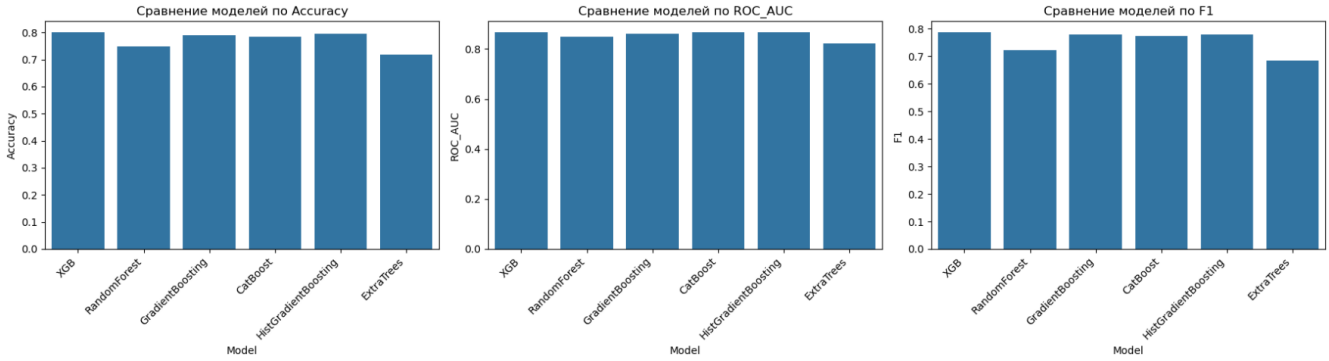

Рисунок 4.1.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 4.1.2.

Таблица 4.1.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	**R2**	**RMSE**	**MAE**
XGB	0.7826	0.7953	0.8656
RandomForest	0.7389	0.7602	0.846
GradientBoosting	0.7636	0.7719	0.8546
CatBoost	0.7799	0.7953	0.8661
HistGradientBoosting	0.7702	0.7836	0.855
ExtraTrees	0.7368	0.7661	0.8637
XGB	0.7826	0.7953	0.8656
RandomForest	0.7389	0.7602	0.846

- Лучшие результаты показали модели CatBoost и XGB — у обеих наивысшие значения Accuracy (0.7953) и ROC AUC (0.8661 и 0.8656 соответственно), а также высокие F1 (0.7799 и 0.7826), что указывает на их устойчивое качество после настройки.
- Худшие показатели у RandomForest и ExtraTrees — наименьшие значения F1 (0.7389 и 0.7368) и Accuracy (0.7602 и 0.7661), хотя ROC AUC остаётся на приемлемом уровне.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 4.1.2.

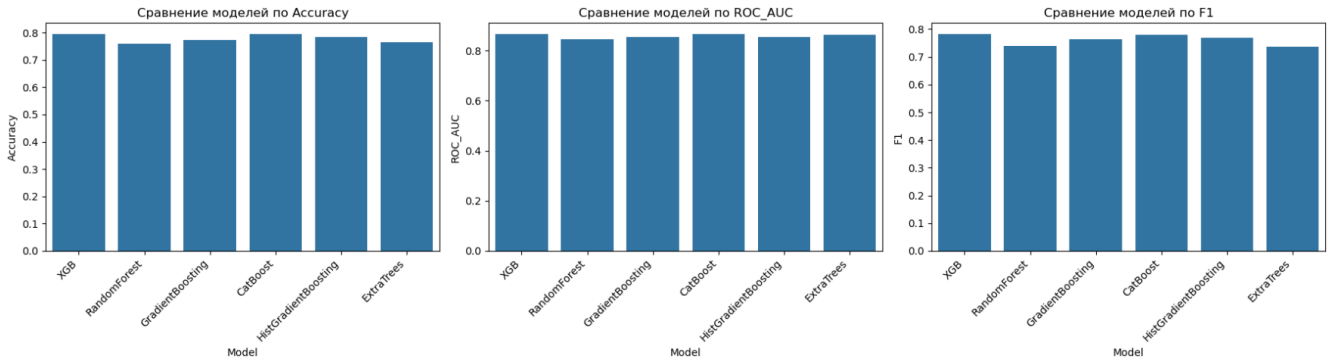

Рисунок 4.1.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 4.1.3):

Таблица 4.1.3 – Общие результаты моделей.

Модель	Accuracy	ROC_AUC	F1	Tuned
XGB	0.8012	0.8671	0.7875	False
CatBoost	0.7953	0.8661	0.7799	True
XGB	0.7953	0.8656	0.7826	True
HistGradientBoosting	0.7953	0.8663	0.7799	False
GradientBoosting	0.7895	0.8618	0.7778	False
CatBoost	0.7836	0.8654	0.773	False
HistGradientBoosting	0.7836	0.855	0.7702	True
GradientBoosting	0.7719	0.8546	0.7636	True
ExtraTrees	0.7661	0.8637	0.7368	True
RandomForest	0.7602	0.846	0.7389	True
RandomForest	0.7485	0.85	0.7226	False

- Лучшие результаты показала модель XGB без подбора — наивысшие значения Accuracy (0.8012), ROC AUC (0.8671) и F1 (0.7875). Очень близкие показатели также у XGB и CatBoost с подбором, а также у HistGradientBoosting без подбора.
- Худшие результаты — у модели ExtraTrees без подбора, с наименьшими значениями всех метрик: Accuracy (0.7193), ROC AUC (0.8229) и F1 (0.6842).

В целом, почти все модели после подбора показали стабильные и высокие результаты, но улучшения по сравнению с лучшей моделью до подбора оказались незначительными. Оптимальной моделью для задачи классификации медианы

IC50 является XGB без подбора, так как она уже демонстрирует наилучшее качество и может применяться даже без дополнительной настройки.

На рисунке 4.1.2 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперапарметров.

Рисунок 4.1.2– Сравнение моделей по метрикам до и после подбора гиперпараметров

4.2 Классификация: превышает ли значение CC50 медианное значение выборки

Сначала проведем тестирование моделей с базовыми настройками, результаты метрик без оптимизации гиперпараметров представлены в таблице 4.2.1.

Таблица 4.2.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	Accuracy	ROC_AUC	F1
XGB	0.8394	0.918	0.8208
RandomForest	0.8031	0.895	0.8041
GradientBoosting	0.8601	0.9271	0.8421
CatBoost	0.8394	0.9237	0.8229
HistGradientBoosting	0.8497	0.9169	0.8362
ExtraTrees	0.7824	0.8416	0.7835

- Лучшие результаты показала модель GradientBoosting — наивысшие значения Accuracy (0.8601), ROC AUC (0.9271) и F1 (0.8421), что свидетельствует о её высокой эффективности даже без настройки.
- Хорошо себя показали также CatBoost и HistGradientBoosting, с близкими значениями метрик.

- Худшие результаты у ExtraTrees — наименьшие значения Accuracy (0.7824), ROC AUC (0.8416) и F1 (0.7835), что говорит о сравнительно слабой способности модели к классификации без настройки.

Визуальное сравнение по метрикам представлено на рисунке 4.2.1.

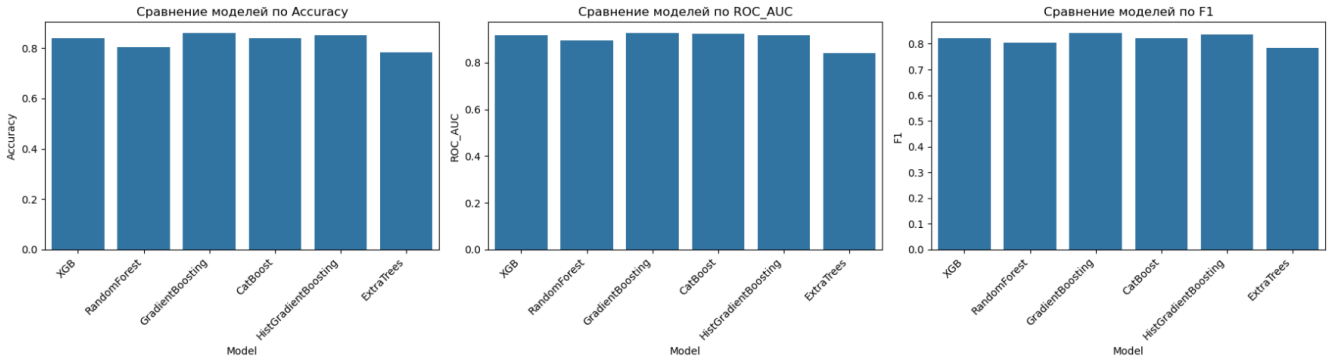

Рисунок 4.2.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 4.2.2.

Таблица 4.2.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	**Accuracy**	**ROC_AUC**	**F1**
XGB	0.8523	0.8653	0.9256
RandomForest	0.8304	0.8497	0.9177
GradientBoosting	0.8249	0.8394	0.9166
CatBoost	0.8242	0.8342	0.9164
HistGradientBoosting	0.8372	0.8549	0.9252
ExtraTrees	0.8256	0.8446	0.9159
XGB	0.8523	0.8653	0.9256
RandomForest	0.8304	0.8497	0.9177

- Лучшие результаты показала модель XGB — наивысшие значения Accuracy (0.8653), ROC AUC (0.9256) и F1 (0.8523), что делает её наиболее сбалансированной и точной после настройки, очень близко к ней по качеству идёт HistGradientBoosting (ROC AUC = 0.9252, F1 = 0.8372, Accuracy = 0.8549).
- Худшие показатели после подбора — у CatBoost, с наименьшей Accuracy (0.8342) и F1 (0.8242), хотя разрыв с другими моделями не критичен.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 4.2.2.

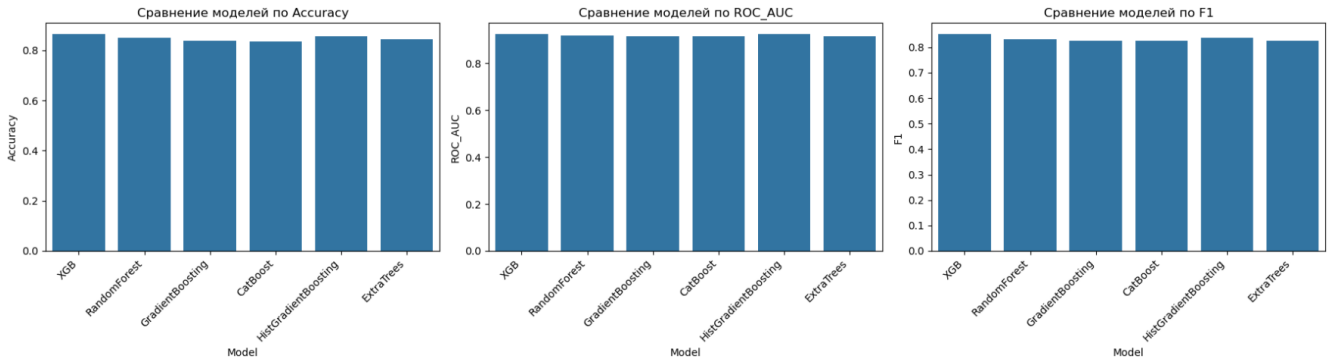

Рисунок 4.2.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 4.2.3):

Таблица 4.2.3 – Общие результаты моделей.

Модель	Accuracy	ROC_AUC	F1	Tuned
XGB	0.8653	0.9256	0.8523	True
GradientBoosting	0.8601	0.9271	0.8421	False
HistGradientBoosting	0.8549	0.9252	0.8372	True
RandomForest	0.8497	0.9177	0.8304	True
HistGradientBoosting	0.8497	0.9169	0.8362	False
ExtraTrees	0.8446	0.9159	0.8256	True
XGB	0.8394	0.918	0.8208	False
CatBoost	0.8394	0.9237	0.8229	False
GradientBoosting	0.8394	0.9166	0.8249	True
CatBoost	0.8342	0.9164	0.8242	True
RandomForest	0.8031	0.895	0.8041	False
ExtraTrees	0.7824	0.8416	0.7835	False

- Лучшие результаты показала модель XGB с подбором — наивысшие значения Accuracy (0.8653), F1 (0.8523), а также почти наивысший ROC AUC (0.9256), что делает её наиболее эффективной и сбалансированной моделью.
- GradientBoosting без подбора тоже показал отличный результат: Accuracy=0.8601, ROC AUC=0.9271, F1=0.8421.

- Худшие результаты — у ExtraTrees без подбора(Accuracy=0.7824, ROC AUC=0.8416, F1 0.7835).

В целом, все модели демонстрируют высокую производительность, однако оптимальной моделью можно считать XGB с подбором гиперпараметров, поскольку она сочетает максимальные параметры, что делает её наилучшим выбором для задачи.

На рисунке 4.2.2 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперапарметров.

Рисунок 4.2.2– Сравнение моделей по метрикам до и после подбора гиперпараметров

4.3 Классификация: превышает ли значение SI медианное значение выборки

Сначала проведем тестирование моделей с базовыми настройками, результаты метрик без оптимизации гиперпараметров представлены в таблице 4.3.1.
Таблица 4.3.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	Accuracy	ROC_AUC	F1
XGB	0.642	0.7233	0.6595
RandomForest	0.6193	0.6791	0.6298
GradientBoosting	0.6364	0.7121	0.6667
CatBoost	0.6477	0.703	0.6667
HistGradientBoosting	0.6591	0.7048	0.6667
ExtraTrees	0.6591	0.7008	0.6552

- Лучшие результаты показала модель HistGradientBoosting и ExtraTrees — обе с самой высокой Accuracy (0.6591) и F1 (0.6667 и 0.6552), а также хорошим ROC AUC (~0.70).

- Худшие показатели у RandomForest — самая низкая Accuracy (0.6193), ROC AUC (0.6791) и F1 (0.6298), что указывает на относительно слабую классификацию без настройки гиперпараметров.

Визуальное сравнение по метрикам представлено на рисунке 4.3.1.

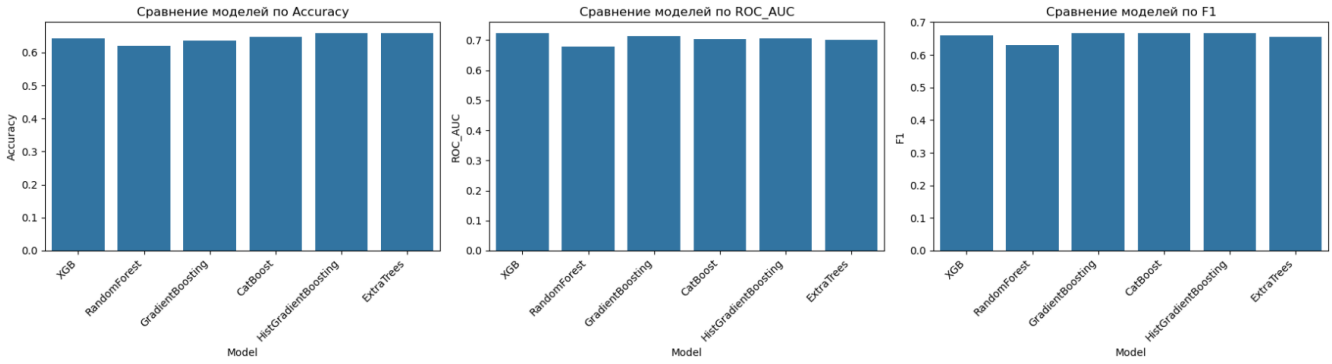

Рисунок 4.3.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 4.3.2.

Таблица 4.3.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	Accuracy	ROC_AUC	F1
XGB	0.6067	0.6023	0.6731
RandomForest	0.6404	0.6364	0.6902
GradientBoosting	0.6484	0.6364	0.6972
CatBoost	0.6629	0.6591	0.6951
HistGradientBoosting	0.6292	0.625	0.7051
ExtraTrees	0.6136	0.6136	0.6638

- Лучшие результаты показала модель CatBoost — наивысшие значения Accuracy (0.6591) и F1 (0.6629), а также высокий ROC AUC (0.6951), что говорит о сбалансированной и точной классификации.
- Худшие показатели у модели XGB — самая низкая Accuracy (0.6023) и F1 (0.6067), несмотря на высокий ROC AUC (0.6731), что может указывать на дисбаланс между precision и recall.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 4.3.2.

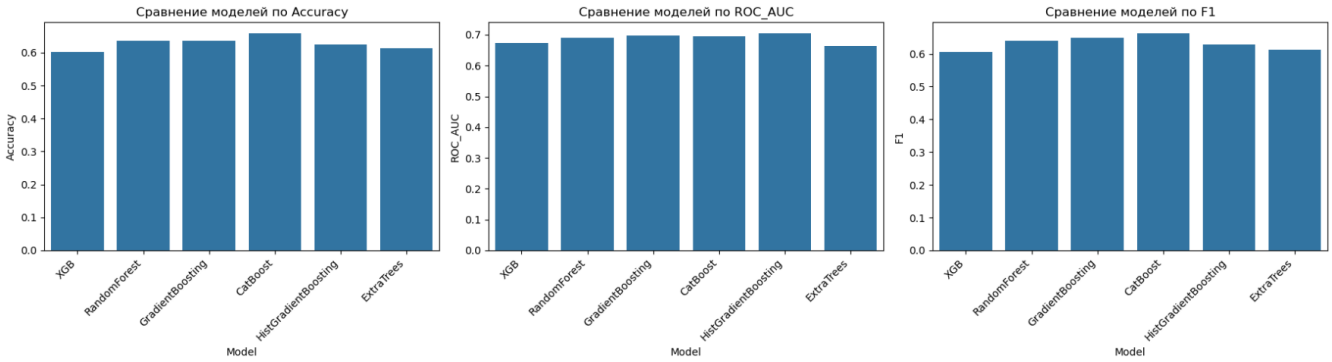

Рисунок 4.3.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 4.3.3):

Таблица 4.3.3 – Общие результаты моделей.

Модель	Accuracy	ROC_AUC	F1	Tuned
ExtraTrees	0.6591	0.7008	0.6552	False
CatBoost	0.6591	0.6951	0.6629	True
HistGradientBoosting	0.6591	0.7048	0.6667	False
CatBoost	0.6477	0.703	0.6667	False
XGB	0.642	0.7233	0.6595	False
GradientBoosting	0.6364	0.7121	0.6667	False
GradientBoosting	0.6364	0.6972	0.6484	True
RandomForest	0.6364	0.6902	0.6404	True
HistGradientBoosting	0.625	0.7051	0.6292	True
RandomForest	0.6193	0.6791	0.6298	False
ExtraTrees	0.6136	0.6638	0.6136	True
XGB	0.6023	0.6731	0.6067	True
ExtraTrees	0.6591	0.7008	0.6552	False
CatBoost	0.6591	0.6951	0.6629	True
HistGradientBoosting	0.6591	0.7048	0.6667	False
CatBoost	0.6477	0.703	0.6667	False

- Лучшие показатели у моделей ExtraTrees без подбора, HistGradientBoosting без подбора и CatBoost с подбором — у всех трёх Accuracy около 0.66, F1 около 0.66, и ROC AUC в диапазоне 0.70.

- Худшие результаты у модели XGB с подбором — самая низкая Accuracy (0.6023) и F1 (0.6067), несмотря на неплохой ROC AUC (0.6731), что говорит о слабом балансе между точностью и полнотой.

В целом, подбор гиперпараметров не всегда улучшает показатели в данной задаче, а оптимальной моделью можно считать CatBoost с подбором, которая демонстрирует стабильный баланс между метриками, либо ExtraTrees без подбора, показывающую схожие хорошие результаты.

На рисунке 4.3.2 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперапарметров.

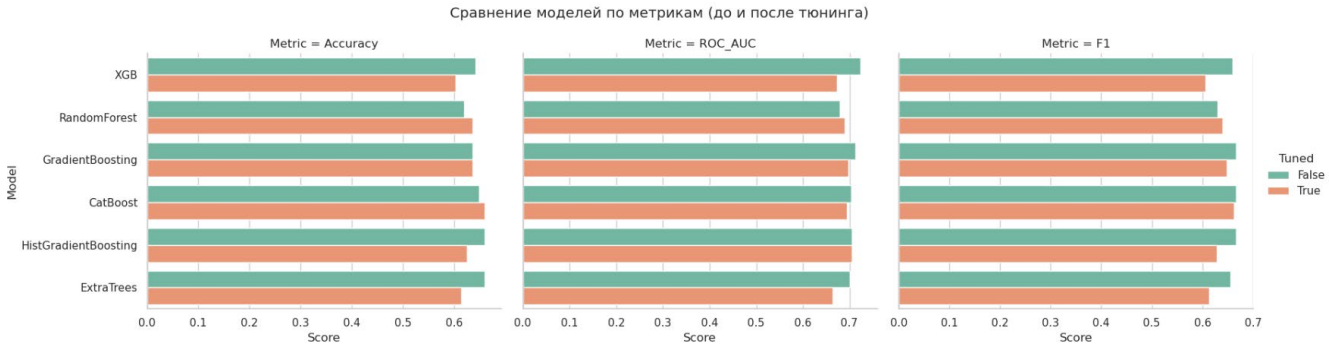

Рисунок 4.3.2– Сравнение моделей по метрикам до и после подбора гиперпараметров

4.4 Классификация: превышает ли значение SI значение 8

При анализе целевой переменной выявлен значительный дисбаланс: класс 0 встречается почти в три раза чаще класса 1 (644 против 232 объектов соответственно). Для устранения этого дисбаланса на тренировочной выборке был применён метод ADASYN, при этом тестовые данные оставались без изменений. В результате после балансировки количество объектов класса 1 увеличилось до 548, а класса 0 — составило 515, тогда как до применения ADASYN на тренировочных данных было 185 объектов класса 1 и 515 класса 0. После этого можно приступать к оценке моделей без подбора гиперпараметров. Результаты метрик без оптимизации гиперпараметров представлены в таблице 4.1.1.
Таблица 4.4.1 – Результаты регрессии без оптимизации гиперпараметров

Модель	Accuracy	ROC_AUC	F1
XGB	0.7102	0.6592	0.4742
RandomForest	0.733	0.6587	0.4946
GradientBoosting	0.7386	0.6367	0.4889

Модель	Accuracy	ROC_AUC	F1
CatBoost	0.75	0.6395	0.4884
HistGradientBoosting	0.7386	0.6346	0.5
ExtraTrees	0.7045	0.6363	0.4694

- Лучшие результаты по Accuracy показала модель CatBoost — 0.75, при этом она демонстрирует F1=0.4884 и ROC AUC=0.6395.
- Также достойные показатели у GradientBoosting и HistGradientBoosting с Accuracy и F1, однако ROC AUC чуть ниже.
- Модели XGB, RandomForest и ExtraTrees показали несколько более низкие метрики, с Accuracy около 0.70–0.73 и F1 около 0.47–0.49, что указывает на необходимость дальнейшего подбора гиперпараметров.

Визуальное сравнение по метрикам представлено на рисунке 4.4.1.

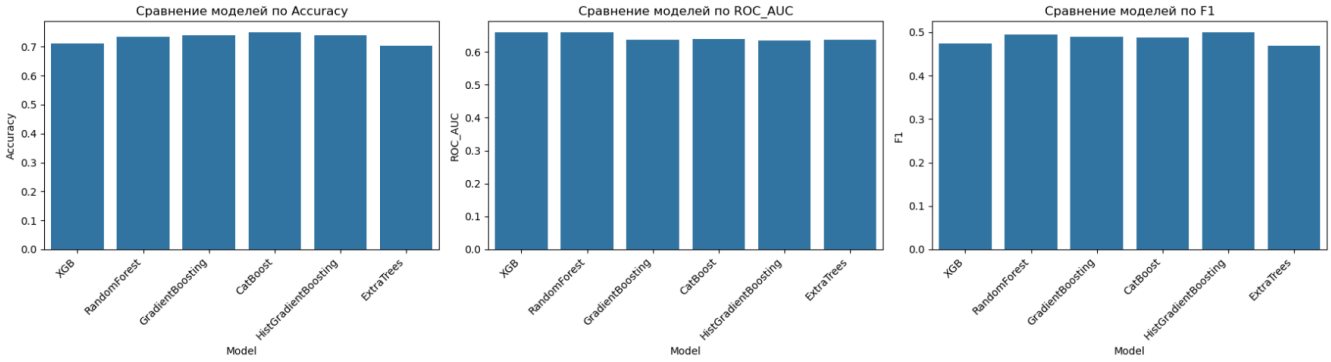

Рисунок 4.4.1– Сравнение моделей по метрикам

Далее проверим метрики после оптимизации, результаты представлены в таблице 4.4.2.

Таблица 4.4.2 – Результаты регрессии после оптимизации гиперпараметров.

Модель	Accuracy	ROC_AUC	F1
XGB	0.4773	0.7386	0.6763
RandomForest	0.4565	0.7159	0.6682
GradientBoosting	0.4719	0.733	0.6395
CatBoost	0.4583	0.7045	0.6168
HistGradientBoosting	0.4419	0.7273	0.6511
ExtraTrees	0.4681	0.7159	0.6357
XGB	0.4773	0.7386	0.6763
RandomForest	0.4565	0.7159	0.6682

- Лучшее Accuracy показала модель XGB (0.7386) с наибольшим ROC AUC (0.6763) и неплохим F1 (0.4773), что делает её самой сбалансированной.
- Худшие показатели у HistGradientBoosting — Accuracy 0.7273, F1 0.4419 и ROC AUC 0.6511, немного ниже остальных.
- В целом, подбор гиперпараметров улучшил Accuracy у XGB и GradientBoosting, но не привёл к значительному росту F1 и ROC AUC для большинства моделей.

Визуальное сравнение по метрикам после подбора гиперпараметров представлено на рисунке 4.4.2.

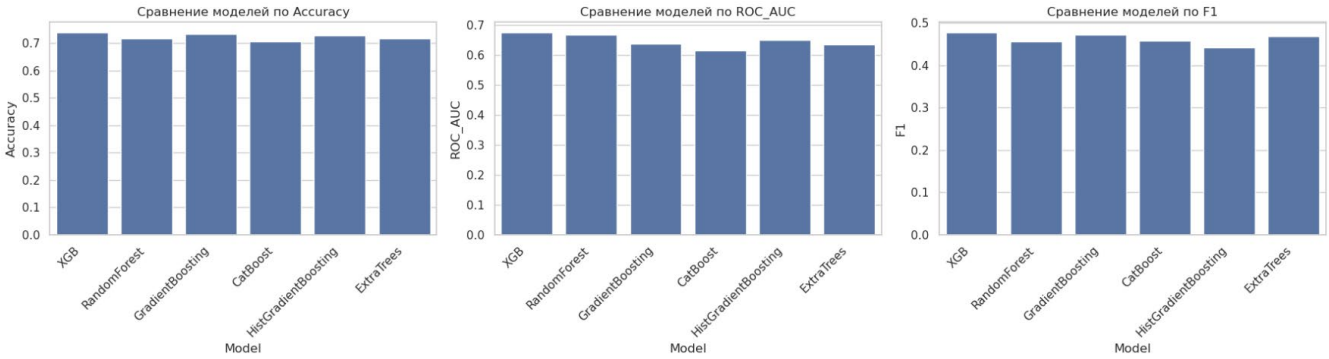

Рисунок 4.4.2– Сравнение моделей по метрикам после подбора гиперпараметров

Далее рассмотрим общую таблицу (Таблица 4.4.3):

Таблица 4.4.3 – Общие результаты моделей.

Модель	Accuracy	ROC_AUC	F1	Tuned
CatBoost	0.75	0.6395	0.4884	False
GradientBoosting	0.7386	0.6367	0.4889	False
XGB	0.7386	0.6763	0.4773	True
HistGradientBoosting	0.7386	0.6346	0.5	False
GradientBoosting	0.733	0.6395	0.4719	True
RandomForest	0.733	0.6587	0.4946	False
HistGradientBoosting	0.7273	0.6511	0.4419	True
ExtraTrees	0.7159	0.6357	0.4681	True
RandomForest	0.7159	0.6682	0.4565	True
XGB	0.7102	0.6592	0.4742	False
ExtraTrees	0.7045	0.6363	0.4694	False
CatBoost	0.7045	0.6168	0.4583	True

Модель	Accuracy	ROC_AUC	F1	Tuned
CatBoost	0.75	0.6395	0.4884	False
GradientBoosting	0.7386	0.6367	0.4889	False
XGB	0.7386	0.6763	0.4773	True
HistGradientBoosting	0.7386	0.6346	0.5	False

- Лучшие показатели Accuracy у модели CatBoost без подбора (0.75), при этом у неё довольно средний ROC AUC (0.6395) и F1 (0.4884).
- Высокая сбалансированность по F1 и ROC AUC у HistGradientBoosting без подбора — F1 = 0.50, ROC AUC = 0.6346.
- Лучшие ROC AUC после подбора показал XGB (0.6763) с хорошей Accuracy (0.7386), хотя F1 немного уступает.
- Худшие результаты у CatBoost с подбором гиперпараметров — по всем метрикам ниже, чем у версии без настройки.

В целом, модель CatBoost без подбора гиперпараметров показывает наилучший баланс точности и полноты для задачи классификации превышения SI > 8. Однако, если важна более высокая ROC AUC, можно рассмотреть XGB с подбором параметров как альтернативу.

На рисунке 4.4.2 представлено сравнение метрик моделей до и после оптимизации, видно, что по большей части метрики улучшаются после подбора гиперапарметров.

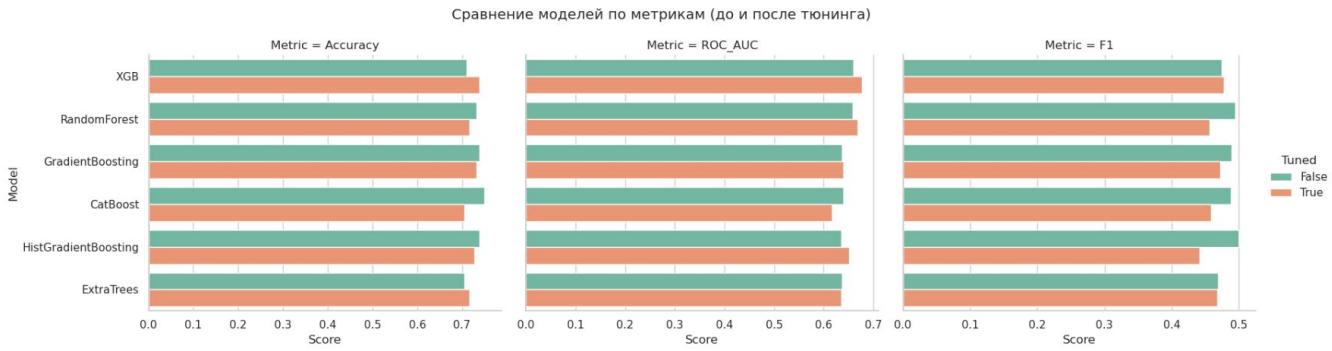

Рисунок 4.4.2– Сравнение моделей по метрикам до и после подбора гиперпараметров

5. Выводы

Исследование охватило 7 задач: 3 регрессионных (прогнозирование IC50, CC50, SI) и 4 классификационных (бинарное разделение по медиане/порогу 8 для IC50, CC50, SI). Для каждой задачи проведён сравнительный анализ моделей до и после оптимизации гиперпараметров с оценкой по метрикам:

Регрессия: R2 (основная), RMSE, MAE.

Классификация: Accuracy (основная), ROC-AUC, F1-score.

5.1. Ключевые результаты

1. Регрессия IC50

Лучшая модель: CatBoost_Tweedie (без оптимизации гиперпараметров, R2=0.3215). Оптимизация не дала улучшений: тюнинг версия показала R2=0.3179. Альтернатива: XGB_Tweedie (R2=0.3023) с более высокими ошибками.

2. Регрессия CC50

Лидер: CatBoost (с оптимизацией гиперпараметров, R2=0.6396). Без оптимизации GradientBoosting (R2=0.6246) близок по R2, но проигрывает по ошибкам.

3. Регрессия SI

Наивысший R2: GradientBoosting (с оптимизацией гиперпараметров, R2=0.1502). Низкие значения R2 у всех моделей указывают на сложность прогнозирования SI.

4. Классификация IC50 (превышение медианы)

Лучшая модель: XGB без оптимизации гиперпараметров (Accuracy=0.8012, ROC-AUC=0.8671). CatBoost с тюнингом показывает Accuracy=0.7953. RandomForest и другие модели уступают по всем метрикам.

5. Классификация CC50 (превышение медианы)

Лидер: XGB с тюнингом (Accuracy=0.8653, ROC-AUC=0.9256). GradientBoosting без оптимизации гиперпараметров имеет ROC-AUC=0.9271, но уступает по Accuracy

6. Классификация SI (превышение медианы)

Наилучшая: HistGradientBoosting без оптимизации гиперпараметров (Accuracy=0.6591, ROC-AUC=0.7048). Некоторые модели показывают такую же Accuracy, но хуже по F1 и ROC-AUC

7. Классификация SI (превышение порога 8)

Лучшая модель: CatBoost без оптимизации гиперпараметров (Accuracy=0.75, ROC-AUC=0.6395). Остальные модели имеют Accuracy 0.733–0.7386

5.2. Общие закономерности

Влияние оптимизации гиперпараметров:

Существенный прирост для регрессии CC50 (CatBoost) и SI (GradientBoosting). Минимальный эффект для IC50 и классификации SI, где лучшие модели — без оптимизации.

Эффективность алгоритмов:

CatBoost и XGBoost доминируют в большинстве задач. GradientBoosting показал лучший результат для SI, но с низким R2.