

# **Национальный исследовательский ядерный университет «МИФИ»**

## **Классическое машинное обучение**

### **Курсовая работа (vo\_RJ)**

#### **Исследование лекарственной активности**

**Студент:**

**Кочетков Михаил Николаевич**



## Введение

### Цель работы

На основе предоставленных данных по химическим соединениям построить прогноз эффективности веществ с целью подбора оптимального состава лекарственного препарата. Основной фокус — на предсказании ключевых показателей активности соединений (IC50, CC50, SI) и их классификации на "сильные" / "слабые" ингибиторы.

### Задачи исследования

1. Провести исследовательский анализ данных (EDA) и оценить информативность признаков.
2. Построить модели машинного обучения:
  - Регрессия:
    - Прогноз значения IC50
    - Прогноз значения CC50
    - Прогноз значения SI
  - Классификация:
    - Бинарный прогноз: превышает ли IC50 медианное значение
    - Бинарный прогноз: превышает ли CC50 медианное значение
    - Бинарный прогноз: превышает ли SI медианное значение
    - Бинарный прогноз: превышает ли SI значение 8
3. Выполнить сравнительный анализ качества моделей по метрикам:
4. Выбрать наиболее эффективные модели и обосновать выбор.
5. Предложить рекомендации по использованию финальной модели в практической работе.

## Глава 1

### Предобработка данных

#### Описание датасета

Датасет представляет собой таблицу, содержащую данные по 1001 химическому соединению. Каждая строка соответствует одному веществу, столбцы — его физико-химическим признакам и биологической активности

#### Признаки:

- **Числовые признаки** : 107 колонок с типом float64
- **Целочисленные признаки** : 107 колонок с типом int64

Признаки описывают структурные, физико-химические и молекулярные свойства соединений.

**Примеры признаков:**

- Молекулярная масса
- Количественные характеристики структуры
- Индикаторы наличия/отсутствия определённых функциональных групп
- Другие числовые и бинарные дескрипторы

Ниже представлена таблице целевых переменных.

Таблица 1. Целевые переменные датасета

IC <sub>50</sub> , mM	Концентрация соединения (в миллимолях), требуемая для подавления вирусной активности на 50%
CC <sub>50</sub> , mM	Концентрация соединения (в миллимолях), вызывающая гибель 50% клеток (цитотоксичность)
SI (Selectivity Index)	Индекс селективности, рассчитываемый как отношение CC <sub>50</sub> к IC <sub>50</sub> (чем выше значение, тем более селективен препарат)

**Полный перечень признаков**

**Электронные и энергетические параметры:**

- **MaxAbsEStateIndex** — максимальный электроотрицательный индекс состояния по абсолютному значению
- **MaxEStateIndex** — максимальный индекс состояния
- **MinAbsEStateIndex** — минимальный электроотрицательный индекс по абсолютному значению
- **MinEStateIndex** — минимальный индекс состояния
- **MaxPartialCharge** — максимальный частичный заряд атома
- **MinPartialCharge** — минимальный частичный заряд атома
- **MaxAbsPartialCharge** — максимальный частичный заряд (по модулю)
- **MinAbsPartialCharge** — минимальный частичный заряд (по модулю)

### Молекулярные дескрипторы:

- **MolWt** — молекулярная масса
- **HeavyAtomMolWt** — масса без учёта атомов водорода
- **ExactMolWt** — точная молекулярная масса
- **NumValenceElectrons** — количество валентных электронов
- **NumRadicalElectrons** — количество радикальных электронов
- **qed** — Quantitative Estimate of Drug-likeness (оценка качества молекулы как кандидата в лекарства)
- **SPS** — сумма поляризационных поверхностей растворителя

### Физико-химические свойства:

- **MolLogP** — коэффициент распределения (оценка липофильности)
- **MolMR** — молярный рефракционный показатель (мера молекулярного объёма и поляризуемости)

### Структурные признаки:

- **HeavyAtomCount** — число тяжёлых атомов (все, кроме H)
- **NHONCount** — число групп OH и NH
- **NOCOUNT** — число атомов N и O
- **NumRotatableBonds** — число ротируемых связей (мера гибкости молекулы)
- **RingCount** — общее число колец
- **FractionCSP3** — доля  $sp^3$ -гибридизованных атомов углерода
- **NumAliphaticRings** — число алифатических колец
- **NumAromaticRings** — число ароматических колец
- **NumHAcceptors** — число акцепторов водородных связей
- **NumHDonors** — число доноров водородных связей
- **NumHeteroatoms** — число гетероатомов (не C/H)

### Дескрипторы Morgan Fingerprint Density:

- **FpDensityMorgan1**, **FpDensityMorgan2**, **FpDensityMorgan3** — плотность фингерпринтов разного радиуса

### BCUT-дескрипторы (атомные свойства):

- **BCUT2D\_MWHI**, **BCUT2D\_MWLOW** — массовые дескрипторы
- **BCUT2D\_CHGHI**, **BCUT2D\_CHGLO** — зарядовые дескрипторы
- **BCUT2D\_LOGPHI**, **BCUT2D\_LOGPLOW** — оценка липофильности

- **BCUT2D\_MRHI, BCUT2D\_MRLOW** — оценка молярного рефракционного индекса

#### Топологические дескрипторы:

- **BalabanJ** — балабановский индекс (топологическая характеристика молекулы)
- **BertzCT** — индекс сложности молекулы (fragment complexity contribution)
- **HallKierAlpha, Ipc, Kappa1, Kappa2, Kappa3** — структурные индексы Холла–Кьера

#### Площадь поверхности доступности (ASA):

- **LabuteASA** — площадь доступной растворителю поверхности

#### PEOE\_VSA — дескрипторы по зарядам:

(PEOE — Partial Equalization of Orbital Electronegativity)

- **PEOE\_VSA1–PEOE\_VSA14** — разделённые по диапазонам значения атомных зарядов и поляризации

#### SMR\_VSA — молекулярное рефракционное значение по участкам:

- **SMR\_VSA1–SMR\_VSA10** — молярная рефракция по различным диапазонам

#### SlogP\_VSA — logP по областям молекулы:

- **SlogP\_VSA1–SlogP\_VSA12** — дескрипторы липофильности по участкам молекулы

#### TPSA — полярная поверхность:

- **TPSA** — суммарная полярная поверхность (Topological Polar Surface Area)

#### EState\_VSA — электроотрицательность по зонам:

- **EState\_VSA1–EState\_VSA11** — деление молекулы на участки по электроотрицательности

#### VSA\_EState — вариация EState по размеру:

- **VSA\_EState1–VSA\_EState9** — деление по электроотрицательности с участием площади поверхности

#### Часто используемые фрагменты (fr ...):

Функциональные группы и их наличие в молекуле:

- **fr\_Al\_COO** — аллильная карбоновая группа
- **fr\_Al\_OH** — спиртовые OH-группы
- **fr\_Al\_OH\_noTert** — OH-группы, за исключением третичных
- **fr\_ArN** — ароматические N
- **fr\_Ar\_COO** — ароматические карбоновые кислоты
- **fr\_Ar\_N** — ароматические амины

- **fr\_Ar\_NH** – ароматические аминогруппы
- **fr\_Ar\_OH** – фенольные OH
- **fr\_COO** – карбоновые кислоты
- **fr\_COO2** – вторая форма карбоновой кислоты
- **fr\_C\_O** – карбонильные группы
- **fr\_C\_O\_noCOO** – карбонилы, кроме карбоновых
- **fr\_C\_S** – группы с атомами C=S
- **fr\_HOCCN** – цианиды с OH-группой
- **fr\_Imine** – имины
- **fr\_NH0** – первичные NH-группы
- **fr\_NH1** – вторичные NH-группы
- **fr\_NH2** – третичные NH-группы
- **fr\_N\_O** – связи N–O
- **fr\_Ndealkylation1, fr\_Ndealkylation2** – маркеры реакции N-деалкилирования
- **fr\_Nhpyrrole** – пиррольные NH-группы
- **fr\_SH** – тиольные группы
- **fr\_aldehyde** – альдегиды
- **fr\_alkyl\_carbamate** – карбаматы
- **fr\_alkyl\_halide** – алкилгалогениды
- **fr\_allylic\_oxid** – метки для окисления аллильных групп
- **fr\_amide** – амиды
- **fr\_amidine** – амидины
- **fr\_aniline** – анилины
- **fr\_aryl\_methyl** – арилметильные группы
- **fr\_azide** – азида
- **fr\_azo** – азо-соединения
- **fr\_barbitur** – барбитуровая кислота или её производные
- **fr\_benzene** – бензольные кольца
- **fr\_benzodiazepine** – бензодиазепиновые структуры
- **fr\_bicyclic** – двухкольцевые структуры
- **fr\_diazo** – диазосоединения
- **fr\_dihydropyridine** – дигидропиридины

- **fr\_epoxide** – эпоксиды
- **fr\_ester** – эфиры
- **fr\_ether** – простые эфиры
- **fr\_furan** – фурановые кольца
- **fr\_guanido** – гуанидиновые группы
- **fr\_halogen** – галогены
- **fr\_hdrzine** – гидразиновые группы
- **fr\_hdrzone** – гидразоны
- **fr\_imidazole** – имидазолы
- **fr\_imide** – имиды
- **fr\_isocyan** – изоцианиды
- **fr\_isothiocyan** – изотиоцианиды
- **fr\_ketone** – кетоны
- **fr\_ketone\_Topliss** – кетоны (по Topliss)
- **fr\_lactam** – лактамы
- **fr\_lactone** – лактоны
- **fr\_methoxy** – метокси-группы
- **fr\_morpholine** – морфолиновые структуры
- **fr\_nitrile** – нитрилы
- **fr\_nitro** – нитрогруппы
- **fr\_nitro\_arom** – нитроароматические соединения
- **fr\_nitro\_arom\_nonortho** – нитроароматические, не орто-замещённые
- **fr\_nitroso** – нитрозо-соединения
- **fr\_oxazole** – оксазолы
- **fr\_oxime** – оксимы
- **fr\_para\_hydroxylation** – метки для пара-гидроксилирования
- **fr\_phenol** – фенольные OH-группы
- **fr\_phenol\_noOrthoHbond** – фенолы без орто-водородных связей
- **fr\_phos\_acid** – фосфорные кислоты
- **fr\_phos\_ester** – фосфорные эфиры
- **fr\_piperdine** – пиперидиновые структуры
- **fr\_piperzine** – пиперазиновые структуры

- **fr\_priamide** – первичные амиды
- **fr\_prisulfonamd** – сульфонамиды
- **fr\_pyridine** – пиридиновые кольца
- **fr\_quatN** – четвертичные атомы азота
- **fr\_sulfide** – сульфиды
- **fr\_sulfonamd** – сульфонамиды
- **fr\_sulfone** – сульфоны
- **fr\_term\_acetylene** – терминальные ацетилены
- **fr\_tetrazole** – тетразолы
- **fr\_thiazole** – тиазолы
- **fr\_thiocyan** – тиоцианаты
- **fr\_thiophene** – тиофеновые кольца
- **fr\_unbrch\_alkane** – неразветвлённые алканы
- **fr\_urea** – мочевины и её производные

## Этапы предобработки

### 1. Загрузка и первичный анализ датасета

Было проведено:

- Вывод списка всех колонок
- Проверка типов данных
- Обзор структуры датасета (head, tail)
- Описательная статистика (describe())

### 2. Удаление нулевых столбцов

Проведён поиск числовых столбцов, где все значения равны нулю. Такие столбцы не несут информационной ценности и были удалены.

Результат:

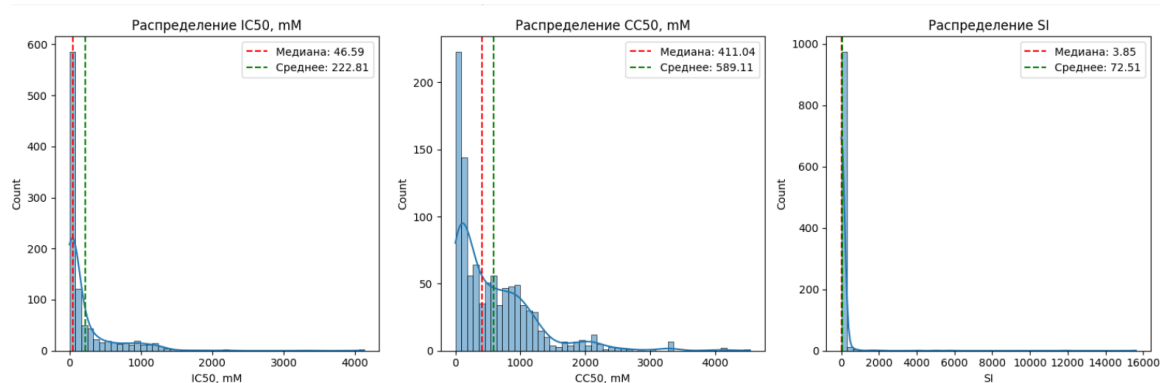
- Найдены и удалены нулевые столбцы.
- Датасет стал менее разреженным.

### 3. Анализ распределения целевых переменных

Гистограммы:

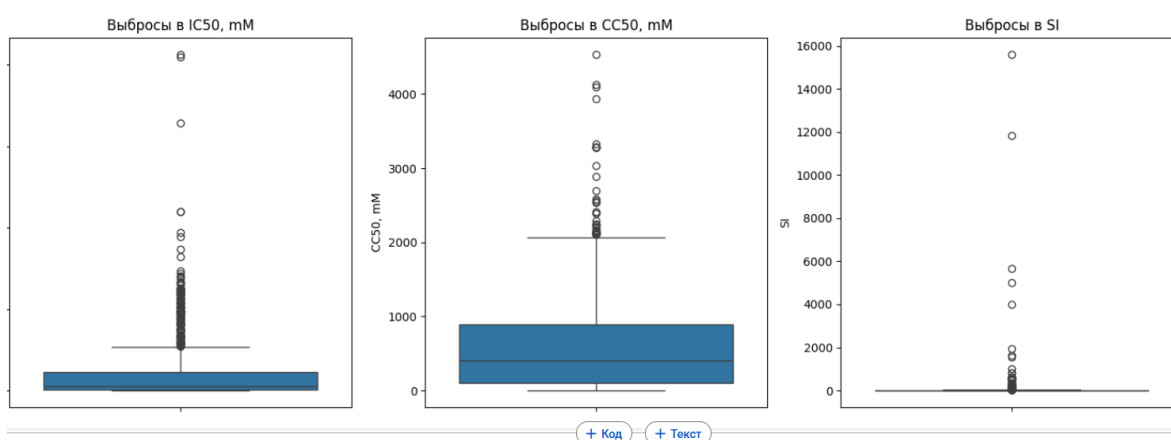
- Построены гистограммы для IC50, CC50, SI.





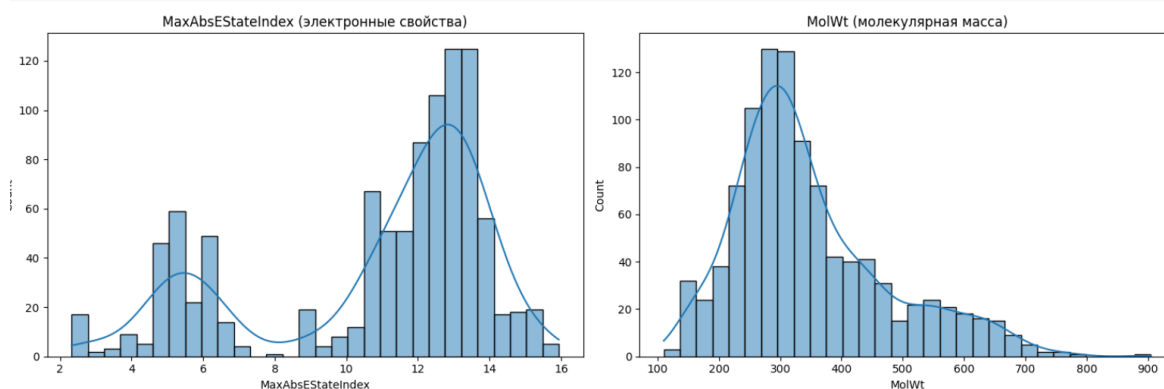
- Для каждой переменной добавлены линии медианы и среднего.

### Boxplot:



- Визуализация выбросов показала:
  - Все три целевые переменные имеют **правостороннюю асимметрию**
  - Выбросы выражены особенно у SI
  - Медиана меньше среднего → влияние хвоста распределения

### 4. Визуализация дескрипторов молекул



**MaxAbsEStateIndex** (электронные свойства): Распределение близко к нормальному с небольшим смещением вправо. Большинство значений сосредоточены между 8 и 15, с редкими выбросами до 16.

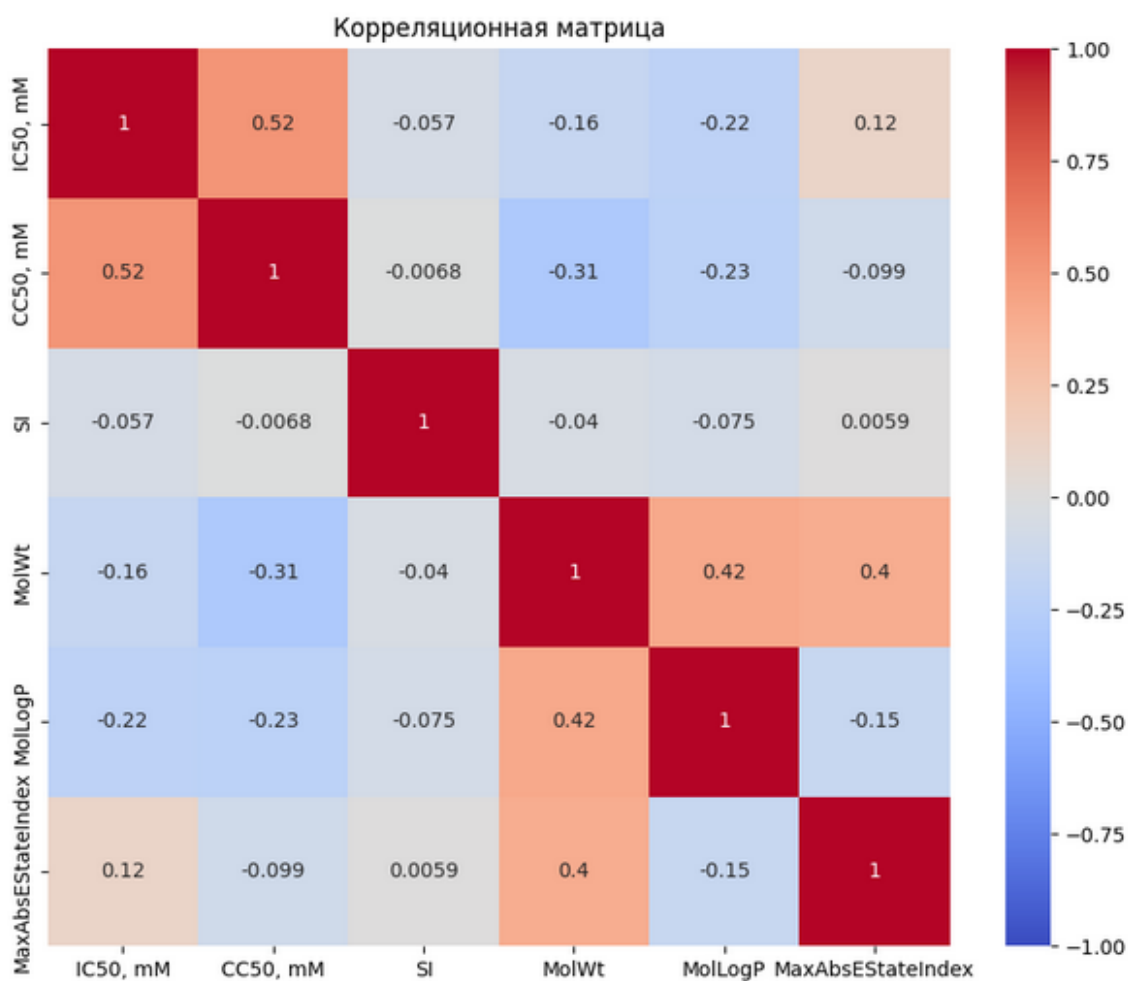
**MolWt (молекулярная масса):** Близко к нормальному распределению с пиком около 300–400. Длинный правый хвост указывает на наличие молекул с высокой массой (>700).

Значение: Для моделей, чувствительных к масштабу (линейные модели), требуется стандартизация/нормализация.

Выбросы в MolWt могут повлиять на регрессию — использовать методы, устойчивые к аномалиям.

## 5. Корреляционный анализ

Малая корреляционная матрица:



Содержит ключевые признаки:

['IC50, mM', 'CC50, mM', 'SI', 'MolWt', 'MolLogP', 'MaxAbsEStateIndex']

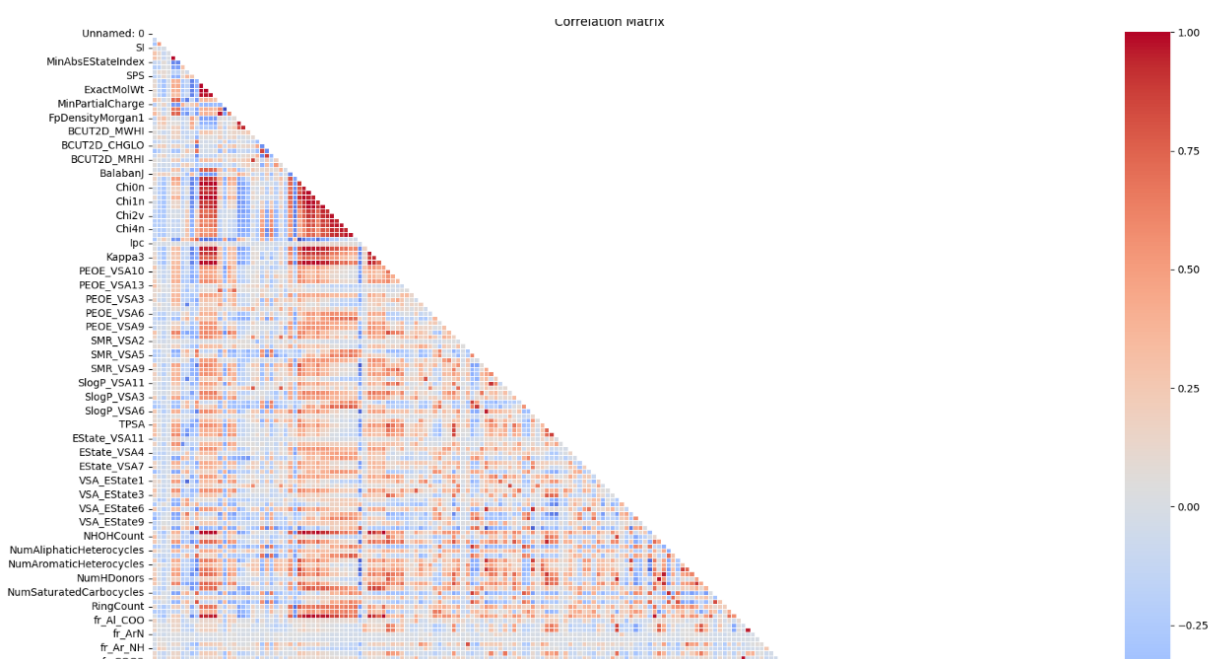
**Ключевые наблюдения:**

- **Высокая корреляция между IC50 и CC50:**  
Коэффициент Пирсона: **~0.52**  
Это указывает на связь между активностью и токсичностью соединений
- **Низкая корреляция SI с другими параметрами :**  
SI почти не связан ни с IC50, ни с CC50  
Подтверждает, что SI — самостоятельный и важный параметр

- **Корреляция MolWt и MolLogP :**  
~0.42  
Логично, так как молекулярная масса влияет на липофильность
- **Корреляция MolLogP и MaxAbsEStateIndex:**  
~0.4  
Указывает на связь между электроотрицательностью атомов и липофильностью молекулы

### Большая корреляционная матрица (все числовые признаки):

*\*матрица содержит большое количество признаков и в читаемом виде может быть представлена только фрагментарно. За подробностями адресую к оригинальному блокноту*



- Показано, что большинство признаков слабо скоррелированы (значения < 0.3)
- Найдены кластеры сильно скоррелированных признаков:
  - Между Chi0 и Chi1 (топологические индексы)
  - Между ExactMolWt и MolWt (~0.999)
  - Chi1 и HeavyAtomCount (~0.998)

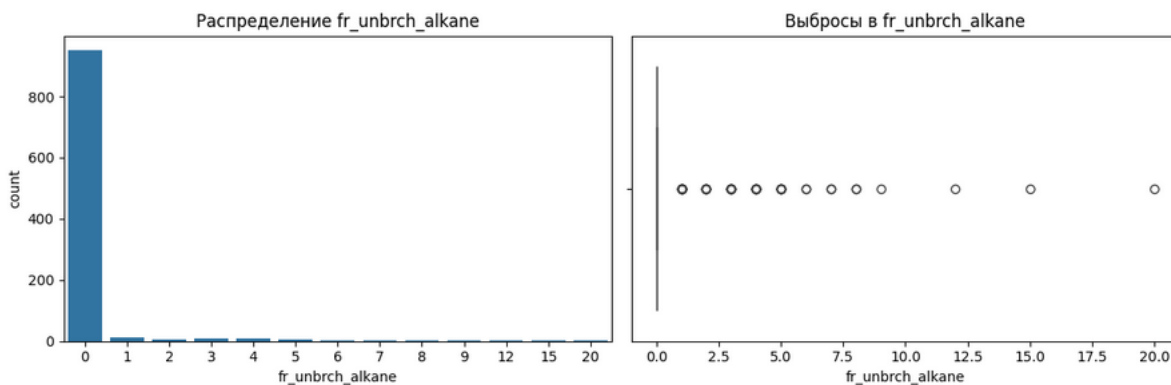
### Проблема мультиколлинеарности:

- Признаки ExactMolWt, Chi0, LabuteASA, Chi0n имеют **очень высокую корреляцию** с другими признаками.
- Эти признаки удалены для снижения влияния мультиколлинеарности.


### Применённые преобразования:

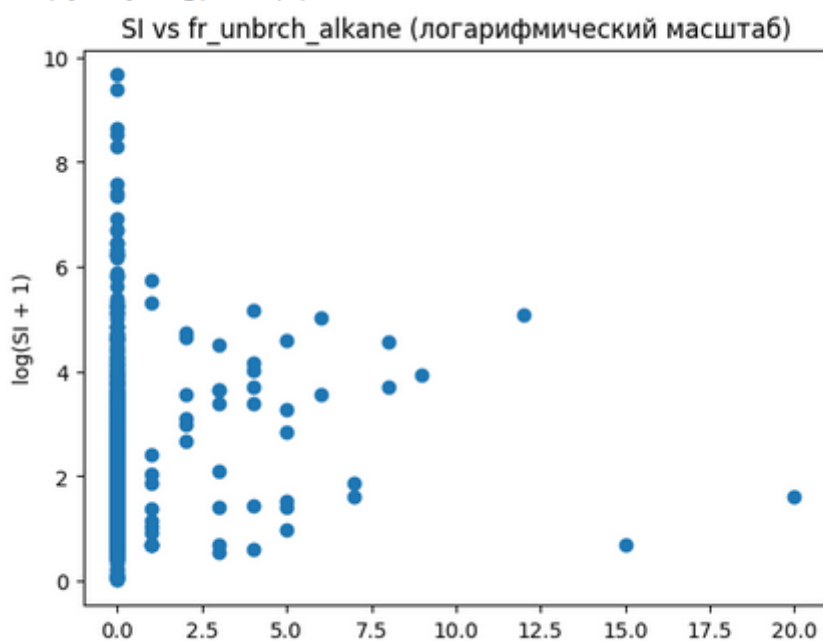
- Удалены сильно коррелирующие признаки
- Для анализа использованы методы регуляризации и проверки на нормальность

## Анализ фрагмента fr\_unbrch\_alkane:



- 99% значений = 0 → низкая вариативность

 `Text(0, 0.5, 'log(SI + 1)')`



- Однако, значение может быть критически важно для прогноза липофильности и биодоступности

## Проверка зависимости с целевыми переменными:

Рассчитаны коэффициенты Спирмена и Пирсона между fr\_unbrch\_alkane и целевыми переменными

Выполнен тест Манна–Уитни для оценки различий между группами (0 и 1 по fr\_unbrch\_alkane)

Не выявлено сильной связи, но признак сохранён из-за возможного влияния на липофильность

Корреляции с fr\_unbrch\_alkane:

IC50, mM:

Спирмен:  $r = -0.0757$ ,  $p = 0.0165$

Пирсон:  $r = -0.0501$ ,  $p = 0.1135$

-

CC50, mM:  
Спирмен:  $r=0.0366$ ,  $p=0.2475$   
Пирсон:  $r=-0.0130$ ,  $p=0.6816$

SI:  
Спирмен:  $r=0.0988$ ,  $p=0.0017$   
Пирсон:  $r=-0.0075$ ,  $p=0.8139$

## Анализ статистик

Анализ статистик в основном выявил особенности целевых переменных и позволяет выделить несколько ключевых моментов:

Общее количество наблюдений: Все переменные имеют 1001 наблюдение, что обеспечивает достаточный объем данных для анализа.

Распределение данных: Скошенность (Skewness): Многие переменные показывают положительную или отрицательную скошенность, что указывает на асимметричность распределения. Например, IC50, mM (скошенность 3.674929) и SI (скошенность 18.013202) имеют значительную положительную скошенность, что может указывать на наличие выбросов или длинного хвоста в правой части распределения. Куртозис (Kurtosis): Значительные значения куртозиса указывают на то, что распределения имеют более выраженные пики по сравнению с нормальным распределением. Например, у Iрс (куртозис 978.251276) наблюдается высокая куртозность, что также может указывать на наличие выбросов.

Основные статистики: Средние значения: Например, среднее значение IC50, mM составляет 222.81, что значительно выше медианного значения (46.59), указывая на наличие высоких значений (выбросов). Стандартное отклонение: Большие значения стандартного отклонения указывают на высокую вариабельность данных. Например, CC50, mM имеет стандартное отклонение 642.87, что говорит о значительных различиях в значениях.

Диапазон значений: Минимальные и максимальные значения показывают широкий диапазон для многих переменных. Например, IC50, mM имеет диапазон от 0.0035 до 4128.53, что указывает на наличие значительных выбросов.

Квартильные значения: Квартильные значения (25%, 50%, 75%) показывают, как распределены данные. Например, для SI 25% значений ниже 1.43, а 75% ниже 16.57, что указывает на значительное количество низких значений.

## Удаление выбросов

Статистика после обработки выбросов:

	IC50, mM	CC50, mM	SI
count	1001.000000	1001.000000	1001.000000
mean	139.705906	589.110728	26.602658
std	224.038972	642.867508	74.625427
min	0.003517	0.700808	0.011489
25%	12.515396	99.999036	1.881818
50%	39.531847	411.039342	5.198095
75%	135.951869	894.089176	24.117647
max	985.642475	4538.976189	828.935484

Выбросы в IC50, mM и SI могли исказить модель. Было сделано:

- Замена значений **IC50 > 1000** на медианное значение
- Замена значений **SI > 1000** на медианное значение

Это позволило:

- Снизить влияние экстремальных значений
- Избежать переобучения на редкие случаи
- Сделать модели более устойчивыми

### **Построение графиков распределения и тесты на нормальность**

После построения графиков распределения и проведения тестов Шапиро-Уилка и Андерсона-Дарлина все колонки были разделены на

- **Колонки имеющие низкую вариативность** - для удаления. Например: SMR\_VSA2', 'SlogP\_VSA7', EState\_VSA1
- **Колонки имеющие распределение отличное от нормального** для преобразования логарифмом. Например: MolWt', 'ExactMolWt', 'NumValenceElectrons'
- **Колонки для преобразования квадратным корнем** – для уменьшения асимметрии 'MaxAbsEStateIndex', 'MaxEStateIndex', 'MinAbsEStateIndex', 'MinEStateIndex',
- **Колонки для бинарного преобразования.** Например 'fr\_Al\_COO' 'fr\_Al\_OH', 'fr\_Al\_OH\_noTert', 'fr\_ArN'

### **Логарифмирование признаков**

Для улучшения распределения и уменьшения влияния длинных хвостов было выполнено:

- **Логарифмирование 43 признаков**, имеющих смещённое распределение
- Использование функции `np.log1p()` ( $\log(1 + x)$ ) для безопасного преобразования

Примеры признаков:

- MolWt, MolMR, BertzCT, Chi1, PEOE\_VSA..., SlogP\_VSA..., EState\_VSA... и другие

После преобразования эти признаки были переименованы с суффиксом `_log`.

### **. Преобразование квадратного корня**

Для части признаков (например, BCUT2D\_MRHI, AvgIpc, BalabanJ) применено преобразование квадратного корня :

- Чтобы уменьшить скошенность распределения
- Для повышения устойчивости моделей к выбросам

Эти признаки были переименованы с суффиксом `_sqrt`.

### **Бинаризация фрагментов**

Все фрагменты (fr\_...) были приведены к бинарному виду:

- Если значение  $> 0 \rightarrow 1$  (признак присутствует)
- Если значение  $== 0 \rightarrow 0$  (признак отсутствует)

Такое преобразование упрощает интерпретацию и уменьшает влияние шума.

### Исследование распределения пропущенных значений:

- Пропуски найдены в следующих признаках:

```
MaxPartialCharge_sqrt      3
MinPartialCharge_sqrt      3
MaxAbsPartialCharge_sqrt   3
MinAbsPartialCharge_sqrt   3
BCUT2D_MWHI_sqrt           3
BCUT2D_MWLOW_sqrt          3
BCUT2D_CHGHI_sqrt          3
BCUT2D_CHGLO_sqrt          3
BCUT2D_LOGPHI_sqrt         3
BCUT2D_LOGPLOW_sqrt        3
BCUT2D_MRHI_sqrt           3
BCUT2D_MRLOW_sqrt          3
dtype: int64
```

- Распределение этих признаков проанализировано через гистограммы и KDE.
- Стратегия заполнения:

После дополнительного анализа распределения по каждой из колонок имеющей пропуски для каждого признака была выбрана своя стратегия заполнения медианой или средним.

```
# Создаем словарь для хранения информации о заполнении
fill_strategy = {
    'MaxPartialCharge_sqrt': 'mean',
    'MinPartialCharge_sqrt': 'median',
    'MaxAbsPartialCharge_sqrt': 'mean',
    'MinAbsPartialCharge_sqrt': 'median',
    'BCUT2D_MWHI_sqrt': 'median',
    'BCUT2D_MWLOW_sqrt': 'median',
    'BCUT2D_CHGHI_sqrt': 'median',
    'BCUT2D_CHGLO_sqrt': 'median',
    'BCUT2D_LOGPHI_sqrt': 'mean',
    'BCUT2D_LOGPLOW_sqrt': 'median',
    'BCUT2D_MRHI_sqrt': 'median',
    'BCUT2D_MRLOW_sqrt': 'median'
}
```

## Масштабирование признаков

Все числовые признаки, кроме целевых переменных, были стандартизированы с помощью StandardScaler.

### Цель:

- Уравнять масштаб признаков
- Подготовить данные для моделей, чувствительных к масштабу (KNN, SVR, XGBoost)

## Сохранение обработанного датасета

Обработанный датасет сохранён в формате CSV:

## Глава 2

### Решение задачи регрессии IC50

**Цель данного этапа** – найти максимально эффективную модель, способную предсказывать значения **IC50**.

Дополнительно важно сравнить между собой:

- **Базовый уровень моделей** (без подбора гиперпараметров),
- **Показатели, достигнутые с помощью GridSearchCV** (полный перебор параметров),
- **Результаты подбора гиперпараметров через Optuna** (байесовская оптимизация).
- **Логарифмирование целевой переменной и повторный подбор параметров через Optuna**

Модели, участвующие в исследовании, представлены в таблице ниже.

Модель	Особенности	Параметры для подбора
<b>Linear Regression</b>	Простая модель, предполагает линейную зависимость между признаками и целевой переменной	fit intercept, normalize (опционально), alpha (если используется Ridge/Lasso)
<b>Random Forest Regressor</b>	Ансамблевая модель на основе множества деревьев, устойчива к переобучению	n_estimators, max_depth, min_samples_split, max_features
<b>Gradient Boosting Regressor</b>	Последовательное построение моделей с корректировкой ошибок	learning_rate, n_estimators, max_depth, subsample
<b>Support Vector Regressor (SVR)</b>	Работает в условиях сложных нелинейных зависимостей	C, epsilon, kernel, gamma
<b>KNeighbors Regressor</b>	Непараметрический метод, основанный на ближайших соседях	n_neighbors, weights, p (метрика расстояния)
<b>XGBoost Regressor</b>	Реализация градиентного бустинга с оптимизацией скорости и памяти	learning_rate, max_depth, n_estimators, subsample, colsample_bytree



Были выбраны стандартные метрики для задачи регрессии

Метрика	Описание	Зачем нужна
<b>RMSE</b> (Root Mean Squared Error)	Среднеквадратичное отклонение предсказанных значений от реальных	Позволяет оценить точность модели с акцентом на большие ошибки; полезна, когда критично минимизировать крупные отклонения в прогнозе активности вещества
<b>MAE</b> (Mean Absolute Error)	Среднее абсолютное отклонение предсказаний от истинных значений	Даёт интуитивно понятную меру средней ошибки модели; устойчива к выбросам и удобна для интерпретации
<b>R<sup>2</sup></b> (Коэффициент детерминации)	Доля дисперсии целевой переменной, объяснённой моделью	Показывает, насколько хорошо модель воспроизводит вариации в данных; позволяет сравнить модель с тривиальным предсказанием среднего значения

Для понимания прогресса моделей проведено обучение без подбора гиперпараметров.

Model	Params	Approach	Time	RMSE	MAE	R <sup>2</sup>
LinearRegression	default	Baseline	-	253.565848	168.921868	-0.182031
RandomForest	default	Baseline	-	218.106676	140.291133	0.125449
GradientBoosting	default	Baseline	-	219.763929	140.942137	0.112108
SVR	default	Baseline	-	256.141315	127.734785	-0.206164
KNeighbors	default	Baseline	-	230.624306	140.264934	0.022184
XGBoost	default	Baseline	-	242.356968	151.975359	-0.079837

По результатам обучения без подбора параметров RandomForest и GradientBoosting показали лучшие результаты "из коробки".

Чтобы сэкономить время приведём сводную таблицу результатов базового уровня, подбора **GridSearchCV** и **Optuna**

Сравнение моделей по метрикам:										
	Model	MAE_Baseline	MAE_GridSearchCV	MAE_Optuna	RMSE_Baseline	RMSE_GridSearchCV	RMSE_Optuna	R <sup>2</sup> _Baseline	R <sup>2</sup> _GridSearchCV	R <sup>2</sup> _Optuna
0	GradientBoosting	140.942137	146.754596	142.842768	219.763929	219.292766	217.463326	0.112108	0.115911	0.130601
1	KNeighbors	140.264934	137.205076	133.854239	230.624306	224.690841	221.984373	0.022184	0.071851	0.094076
2	LinearRegression	168.921868	NaN	NaN	253.565848	NaN	NaN	-0.182031	NaN	NaN
3	RandomForest	140.291133	144.080745	141.829151	218.106676	217.892849	215.862792	0.125449	0.127163	0.143351
4	SVR	127.734785	129.826904	129.826907	256.141315	232.644343	232.644352	-0.206164	0.004979	0.004979
5	XGBoost	151.975359	143.404539	131.589829	242.356968	217.728497	212.310020	-0.079837	0.128479	0.171318

Baseline иногда близок к оптимальному

Для моделей **SVR** и **KNeighbors** подбор гиперпараметров не привёл к существенному улучшению качества. Это может говорить о том, что либо данные имеют низкую информативность признаков, либо модель уже была достаточно устойчивой и адаптированной "из коробки".

Для большинства моделей (XGBoost, RandomForest, GradientBoosting) метод Optuna даёт наибольшее значение  $R^2$  и наименьшие значения RMSE/MAE.

GridSearchCV работает не хуже, но и не лучше Optuna

В некоторых случаях (например, RandomForest) он показывает сопоставимые с Optuna результаты. **Однако в среднем Optuna позволяет достигать лучших метрик за меньшее число итераций, что делает его более эффективным инструментом подбора гиперпараметров.**

Некоторые модели работают стабильно хуже других

Модели LinearRegression и SVR показывают низкое качество прогноза, особенно по метрике  $R^2$ . Эти модели, видимо, недостаточно подходят для данной задачи без дополнительной работы над признаками или преобразования данных.

Исследование показало, что наилучшее качество прогнозирования IC50 достигается при использовании модели XGBoost с гиперпараметрами, подобранными с помощью Optuna . Полученная модель демонстрирует  $R^2 = 0.171$  , что является наилучшим результатом среди всех исследованных моделей и методов настройки .

Однако, следует учитывать, что:

Значение  $R^2 = 0.171$  свидетельствует о умеренной способности модели объяснять вариацию целевой переменной. Это может говорить как о сложности самой задачи, так и о необходимости дальнейшей работы над качеством признаков, их преобразованием или расширением.

Напомним, что на этапе предобработки мы провели логарифмирование для группы признаков, но не стали проводить его для целевых переменных. Все результаты выше показаны на данных без логарифмического преобразования.

### Логарифмирование целевой переменной

Целевая переменная была логарифмирована для уменьшения влияния возможных выбросов:

$$y_{log} = \log(1 + IC50)$$

Обучение проводилось на логарифме, оценка метрик — в логарифмированном пространстве. Модель RMSE MAE  $R^2$  XGBoost 1.367 1.099 0.336 RandomForest 1.376 1.085 0.327 GradientBoosting 1.395 1.144 0.309 KNeighbors 1.445 1.127 0.258 SVR 1.499 1.121 0.201

Вывод: Логарифмирование значительно улучшило метрики. XGBoost объяснила около 34% дисперсии в логарифмированной

После полного цикла исследовательского анализа и оптимизации гиперпараметров с помощью Optuna:

Лучшей моделью стала XGBoost , обученная на логарифме целевой переменной . Она достигла значения  $R^2 = 0.336$  , что является наилучшим результатом среди всех протестированных подходов. **Это говорит о том, что модель способна объяснить около 34% вариации значения IC50 , что может быть достаточным для ранжирования соединений по активности, но недостаточно для точного прогноза.** Дальнейшее

улучшение качества возможно только через добавление новых признаков либо через переход к классификации ("сильный/слабый ингибитор").

## Обратное преобразование прогноза к оригинальной шкале

Для перехода к исходным значениям использовано: python

```
y_pred_original = np.expml(y_pred_log)
```

Модель Метод RMSE MAE R<sup>2</sup> XGBoost Optuna + expml(log\_IC50) 236.20 120.48 -0.03

Вывод: После обратного преобразования качество резко упало.  $R^2 < 0$  говорит о том, что модель работает хуже константы. Это указывает на ограниченную информативность признаков в текущем виде.

№	Модель	Параметры	Подход	Время, сек	RMSE	MAE	R <sup>2</sup>
0	XGBoost	{'learning_rate': 0.041974862047979455, 'max_depth': 6, 'n_estimators': 172, 'subsample': 0.9}	Optuna (log)	38.90	1.367	1.099	0.336
1	RandomForest	{'n_estimators': 50, 'max_depth': 10, 'min_samples_split': 2}	Optuna (log)	108.06	1.376	1.085	0.327
2	GradientBoosting	{'learning_rate': 0.05738659139805542, 'n_estimators': 100, 'max_depth': 5, 'min_samples_split': 5}	Optuna (log)	97.71	1.395	1.144	0.309
3	KNeighbors	{'n_neighbors': 5, 'weights': 'uniform', 'p': 1}	Optuna (log)	1.60	1.445	1.127	0.258
4	SVR	{'C': 10, 'epsilon': 0.199921728605941, 'kernel': 'rbf'}	Optuna (log)	65.35	1.499	1.121	0.201

## Лучшие параметры XGBoost:

- **learning\_rate:** 0.042 (скорость обучения)
- **max\_depth:** 6 (максимальная глубина деревьев)
- **n\_estimators:** 172 (количество деревьев в ансамбле)
- **subsample:** 0.9 (доля случайных образцов для каждого дерева)

Еще раз подчеркнём общий вывод. **Лучшая модель XGBoost не позволяет точно предсказывать значения IC50, но дает возможность провести ранжирование соединений по их активности и использовать результат регрессии как основу для дальнейшего анализа при переходе от прогноза к классификации ("сильный/слабый ингибитор").**

## Глава 3

### Решение задачи регрессии для CC50

В рамках данного исследования предпринята попытка улучшить качество прогнозирования на основе имеющихся данных за счёт применения различных подходов к преобразованию целевой переменной и автоматическому отбору наиболее значимых признаков .

В качестве базового уровня (baseline) выбрано логарифмическое преобразование целевой переменной CC50, mM ( $\log(1 + x)$ ) , что позволило снизить правостороннюю асимметрию распределения и повысить устойчивость моделей к выбросам. На этом наборе данных проводится первая серия экспериментов: обучение моделей без отбора признаков и с автоматическим отбором важных фич .

Дальнейшие эксперименты будут проведены с использованием альтернативного преобразования — Yeo-Johnson , которое также направлено на нормализацию распределения, но в отличие от логарифмирования, корректно обрабатывает нулевые и отрицательные значения. Эти результаты позволят сравнить эффективность двух подходов

к трансформации данных. Для подбора гиперпараметров во всех случаях будет использоваться OPTUNA

На основании результатов предварительных экспериментов был скорректирован набор используемых моделей. Из дальнейшего анализа исключены следующие алгоритмы:

Линейная регрессия — показала недостаточно высокое качество на нелинейных данных;  
Support Vector Regressor (SVR) — стабильно проигрывала по метрикам другим моделям и требовала значительных вычислительных ресурсов;

KNeighborsRegressor — исключена из-

за отсутствия встроенной оценки важности признаков и сложностей с автоматическим отбором фич, что затрудняет интеграцию в pipeline.

Обновлённый список моделей

Все выбранные модели поддерживают механизм оценки важности признаков, что позволяет использовать их в связке с методами автоматического отбора (например, SelectFromModel):

Модель	Краткое описание
<b><i>Random Forest Regressor</i></b>	Ансамблевый метод на основе множества деревьев решений. Устойчив к переобучению, не требует тонкой настройки и хорошо подходит для начального анализа.
<b><i>Gradient Boosting Regressor</i></b>	Последовательный ансамблевый метод, строящий модели с учётом ошибок предыдущих. Обладает хорошей предсказательной способностью, но может быть чувствителен к настройке гиперпараметров.
<b><i>HistGradientBoostingRegressor</i></b>	Усовершенствованная реализация градиентного бустинга от библиотеки scikit-learn. Быстро обучается, поддерживает пропуски и работает эффективно даже на больших данных.
<b><i>XGBoost Regressor</i></b>	Высокоэффективная реализация градиентного бустинга с оптимизацией по скорости и качеству. Хорошо показывает себя на сложных задачах регрессии, поддерживает регуляризацию и параллельные вычисления.
<b><i>LGBMRegressor (LightGBM)</i></b>	Высокопроизводительная реализация градиентного бустинга от Microsoft. Отличается быстрым обучением и низким потреблением памяти. Эффективна при большом числе признаков.
<b><i>CatBoostRegressor</i></b>	Реализация градиентного бустинга от Яндекса. Отлично обрабатывает числовые и категориальные данные, имеет встроенную защиту от переобучения и стабильную сходимость.

Эти модели демонстрируют высокую эффективность на числовых данных с нелинейными зависимостями и хорошо зарекомендовали себя в задачах регрессии как в академической, так и в прикладной практике.

Набор метрик остается прежним

Метрика	Описание	Зачем нужна
<b><i>RMSE (Root Mean Squared Error)</i></b>	Среднеквадратичное отклонение предсказанных значений от реальных	Позволяет оценить точность модели с акцентом на большие ошибки; полезна, когда критично минимизировать крупные отклонения в прогнозе активности вещества
<b><i>MAE (Mean Absolute Error)</i></b>	Среднее абсолютное отклонение предсказаний от истинных значений	Даёт интуитивно понятную меру средней ошибки модели; устойчива к выбросам и удобна для интерпретации
<b><i>R<sup>2</sup> (Коэффициент детерминации)</i></b>	Доля дисперсии целевой переменной, объяснённой моделью	Показывает, насколько хорошо модель воспроизводит вариации в данных; позволяет сравнить модель с тривиальным предсказанием среднего значения

Для каждой из шести моделей проведено обучение на четырёх этапах:

- Baseline : обучение без подбора параметров
- Optuna : подбор гиперпараметров через библиотеку Optuna
- Optuna + Selection : с автоматическим отбором важных фич
- Johnson + Selection : Yeo-Johnson + отбор признаков

Сводная таблица по всем этапам

Stage	MAE				R2				RMSE			
	Baseline	Johnson + Selection	Optuna	Optuna + Selection	Baseline	Johnson + Selection	Optuna	Optuna + Selection	Baseline	Johnson + Selection	Optuna	Optuna + Selection
Model												
CatBoost	0.795100	0.508600	0.796900	0.814800	0.427100	0.513300	0.429800	0.458300	1.141500	0.701600	1.138800	1.110000
GradientBoosting	0.816000	0.560500	0.803200	0.861600	0.410900	0.491200	0.429800	0.432800	1.157500	0.717300	1.138800	1.135800
HistGradientBoosting	0.814200	0.530400	0.822700	0.812300	0.386100	0.488100	0.400700	0.412400	1.181600	0.719500	1.167500	1.156000
LGBM	0.815900	0.530100	0.826100	0.839700	0.389100	0.473300	0.414900	0.390300	1.178700	0.729900	1.153600	1.177500
RandomForest	0.810800	0.523300	0.810200	0.815000	0.437900	0.510000	0.443400	0.431400	1.130600	0.704000	1.125200	1.137200
XGBoost	0.801600	0.533200	0.857900	0.798800	0.418300	0.477800	0.449300	0.437700	1.150200	0.726700	1.119100	1.130800

Анализ результатов

Лучшая модель: CatBoost

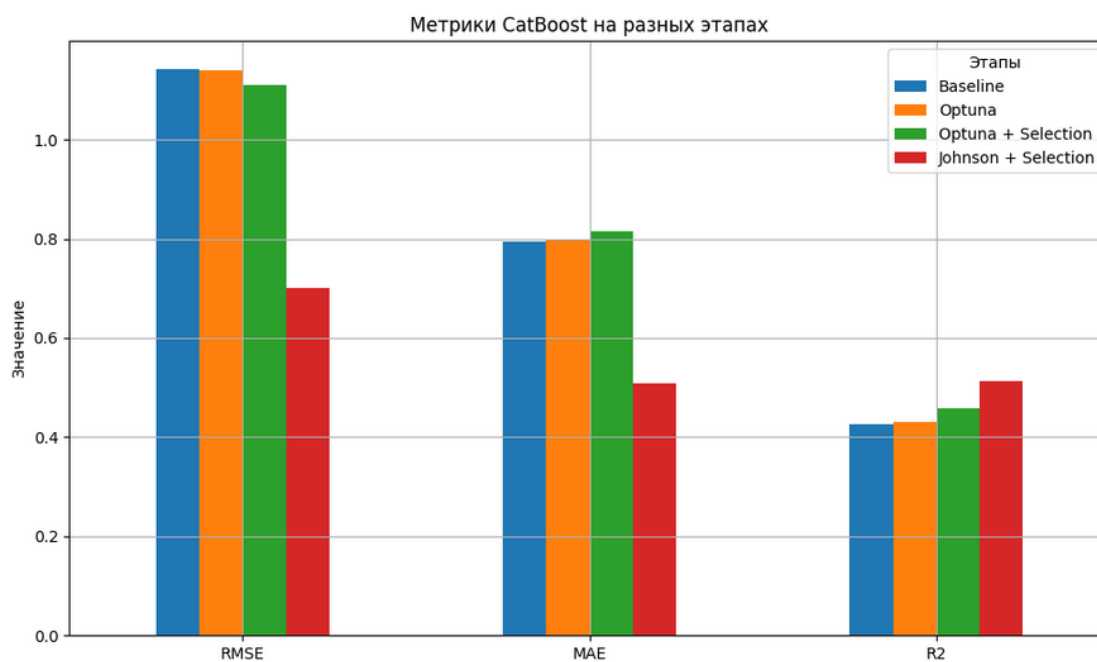
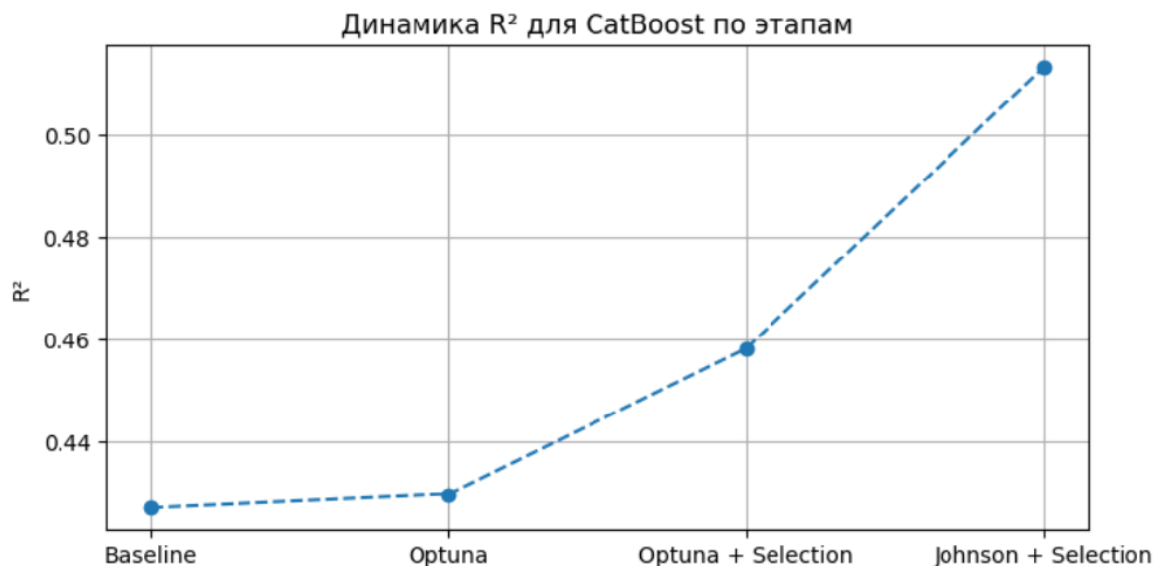
Наибольшее улучшение показала модель CatBoost после применения Yeo-Johnson преобразования и автоматического отбора признаков:

RMSE = 0.7016

MAE = 0.5086

$R^2 = 0.5133$

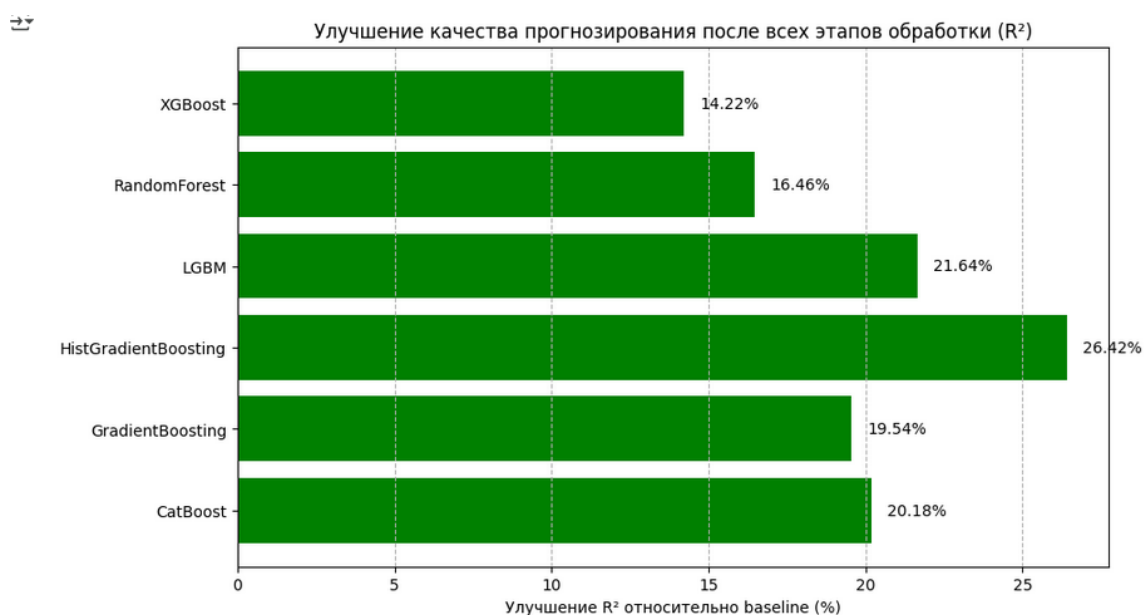
Это говорит о том, что модель объяснила более 51% дисперсии целевой переменной, что является наилучшим результатом среди всех протестированных подходов.



#### 4Влияние Yeo-Johnson

Преобразование целевой переменной методом Yeo-Johnson дало значительное улучшение для большинства моделей:

	Model	R <sup>2</sup> до Johnson	R <sup>2</sup> после Johnson	Улучшение (%)
0	CatBoost	0.427100	0.513300	20.180000
1	RandomForest	0.437900	0.510000	16.460000
2	HistGradientBoosting	0.386100	0.488100	26.420000
3	XGBoost	0.418300	0.477800	14.220000
4	GradientBoosting	0.410900	0.491200	19.540000
5	LGBM	0.389100	0.473300	21.640000



Сокращение сложности:

Отбор признаков позволил уменьшить число используемых фич с 214 до 93, сохранив высокое качество прогнозирования.

Вывод: Модель **CatBoost** показала высокую корректность в ранговом сравнении веществ по значению **CC50**, **mM**.

На основе полученных результатов она может быть рекомендована для решения задачи **ранжирования химических соединений по уровню токсичности**.

Такой подход позволяет:

- Проводить **эффективную сортировку** соединений от "наименее токсичных" к "наиболее токсичным"
- Снижать зависимость от **высокой точности абсолютных значений**, что особенно важно при работе с шумными или слабо предсказуемыми данными
- Опирается на **относительные различия** между веществами, которые модель воспроизводит стабильно и с высокой степенью согласованности

# Глава 4

## Регрессия SI

С учетом ранее проделанной работы позволим себе сразу реализовать метод который показал наибольшую эффективность: Для выбранного списка моделей провести подбор гиперпараметров при помощи Optuna, преобразование целевой переменной Yeo-Johnson и автоматический отбор признаков влияющих на целевую переменную, в данном случае SI

Модель	Краткое описание
Random Forest Regressor	Ансамблевый метод на основе множества деревьев решений. Устойчив к переобучению, не требует тонкой настройки и хорошо подходит для начального анализа.
Gradient Boosting Regressor	Последовательный ансамблевый метод, строящий модели с учётом ошибок предыдущих. Обладает хорошей предсказательной способностью, но может быть чувствителен к настройке гиперпараметров.
HistGradientBoostingRegressor	Усовершенствованная реализация градиентного бустинга от библиотеки <code>skikit-learn</code> . Быстро обучается, поддерживает пропуски и работает эффективно даже на больших данных.
XGBoost Regressor	Высокоэффективная реализация градиентного бустинга с оптимизацией по скорости и качеству. Хорошо показывает себя на сложных задачах регрессии, поддерживает регуляризацию и параллельные вычисления.
LGBMRegressor (LightGBM)	Высокопроизводительная реализация градиентного бустинга от Microsoft. Отличается быстрым обучением и низким потреблением памяти. Эффективна при большом числе признаков.
CatBoostRegressor	Реализация градиентного бустинга от Яндекса. Отлично обрабатывает числовые и категориальные данные, имеет встроенную защиту от переобучения и стабильную сходимость.

Все выбранные модели поддерживают механизм оценки важности признаков, что позволяет использовать их в связке с методами автоматического отбора (например, `SelectFromModel`)

Метрика	Описание	Зачем нужна
RMSE (Root Mean Squared Error)	Среднеквадратичное отклонение предсказанных значений от реальных	Позволяет оценить точность модели с акцентом на большие ошибки; полезна, когда критично минимизировать крупные отклонения в прогнозе активности вещества
MAE (Mean Absolute Error)	Среднее абсолютное отклонение предсказаний от истинных значений	Дает интуитивно понятную меру средней ошибки модели; устойчива к выбросам и удобна для интерпретации
R <sup>2</sup> (Коэффициент детерминации)	Доля дисперсии целевой переменной, объяснённой моделью	Показывает, насколько хорошо модель воспроизводит вариации в данных; позволяет сравнить модель с тривиальным предсказанием среднего значения

Использованы стандартные параметры `trial 30`

Для отбора важных признаков использована модель случайного леса

```
feature_selector_model = RandomForestRegressor(n_estimators=100, random_state=42)
```

Выбрано 93 признаков из 185

Пространство гиперпараметров для каждой модели определено в словаре.

С аналогичным словарем мы работали для решения задачи регрессии для других целевых переменных

Таблица пространства гиперпараметров моделей

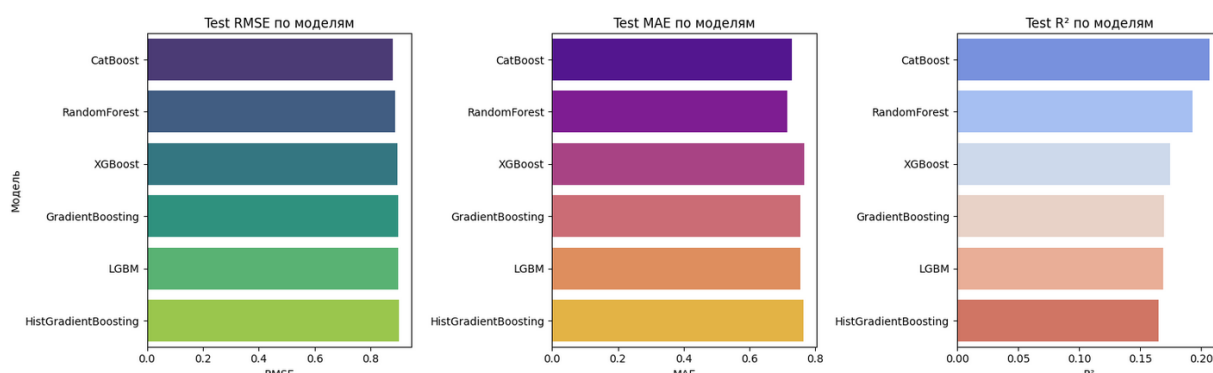
Модель	Гиперпараметр	Возможные значения
RandomForest	n_estimators	[50, 100, 200]
	max_depth	[None, 5, 10]
	min_samples_split	[2, 5]
GradientBoosting	learning_rate	[0.01, 0.1, 0.2]
	n_estimators	[50, 100, 200]
	max_depth	[3, 5, 7]
HistGradientBoosting	learning_rate	[0.01, 0.1, 0.2]
	max_depth	[3, 5, 7, None]
	l2_regularization	[0.0, 1.0, 10.0]
XGBoost	learning_rate	[0.01, 0.1, 0.2]
	max_depth	[3, 5, 7]
	n_estimators	[50, 100, 200]
LGBM	subsample	[0.8, 1.0]
	learning_rate	[0.01, 0.1, 0.2]
	num_leaves	[31, 63, 127]
CatBoost	max_depth	[3, 5, 7]
	n_estimators	[50, 100, 200]
	learning_rate	[0.01, 0.1, 0.2]
CatBoost	depth	[3, 5, 7]
	n_estimators	[50, 100, 200]
	l2_leaf_reg	[1, 3, 5]



Ниже представлены результаты моделей

Результаты Johnson + Optuna с дополнительным отбором признаков:

	Model	Test RMSE	Test MAE	Test R <sup>2</sup>
0	CatBoost	0.878764	0.729825	0.206456
1	RandomForest	0.886357	0.714508	0.192683
2	XGBoost	0.896278	0.766495	0.174511
3	GradientBoosting	0.899150	0.754308	0.169212
4	LGBM	0.899415	0.754696	0.168721



Прогнозирование значения SI (selectivity index) оказалось более сложной задачей, чем прогнозирование CC50 или IC50.

Все протестированные модели показали умеренную эффективность:

Лучшая модель: CatBoost ( $R^2 = 0.20645$ ,  $RMSE = 0.878764$ )

Остальные модели демонстрируют схожие метрики

Возможные причины низкого качества:

SI рассчитывается как отношение  $CC50 / IC50$ , и любые погрешности в прогнозе одной из этих величин усиливаются

Слабая связь между фичами и конечной целевой переменной

Шум и выбросы в SI

Таким образом, использование регрессионных моделей для прогнозирования точного значения SI малопригодно, однако модель может быть использована для ранжирования веществ по уровню селективности.

## Глава 5

### Классификация $IC50 >$ медианы

Для перехода к задаче классификации потребовались преобразования включающие вычисление медианы и присвоение всем строкам метрики по у в соответствии с тем превышает ли значение IC50 медиану.

```
[ ] # Вычисление медианы
median_ic50 = y.median()

[ ] # Преобразование в задачу классификации: IC50 > медиана ?
y_class = (y > median_ic50).astype(int)
```

Использовались метрики для классификации

Метрика	Описание	Зачем нужна
Accuracy	Доля правильных предсказаний среди общего числа	Общая мера эффективности модели; удобна при сбалансированных классах
Precision (Точность)	Доля верно предсказанных положительных объектов среди всех предсказанных положительных	Важна, когда ложные срабатывания критичны (например, дорого проверять вещества в реальности)
Recall (Полнота)	Доля верно предсказанных положительных объектов среди всех реальных положительных	Критична, если важно найти как можно больше истинных случаев (например, пропустить активное вещество нельзя)
F1 Score	Среднее гармоническое между Precision и Recall	Хорошая обобщающая метрика, особенно при дисбалансе классов
ROC AUC	Площадь под ROC-кривой; отражает способность модели отличать классы при разных порогах	Показывает, насколько модель уверенно ранжирует объекты; полезна для сравнения моделей

и уже известный набор моделей.

Модель	Библиотека	Классификатор / Регрессор	Основные преимущества
Random Forest	sklearn.ensemble	RandomForestClassifier	Устойчивость к переобучению, интерпретируемость
HistGradientBoosting	sklearn.ensemble	HistGradientBoostingClassifier	Поддержка NaN, быстрое обучение
XGBoost	xgboost	XGBClassifier	Высокая точность, мощный тюнинг гиперпараметров
LightGBM (LGBM)	lightgbm	LGBMClassifier	Высокая скорость, эффективное дерево-рост
CatBoost	catboost	CatBoostClassifier	Отличная стабильность, автоматическая обработка числовых признаков

обоснование выбора:

- Все модели поддерживают гибкий тюнинг гиперпараметров.
  - Все они умеют автоматически отбирать важные признаки.
  - Они хорошо зарекомендовали себя в научных задачах (в том числе в хемоинформатике).
- Классы предсказуемо распределены почти поровну

```
Частота классов:
IC50, mM
0    532
1    469
Name: count, dtype: int64

Доли классов:
IC50, mM
0    0.531469
1    0.468531
Name: proportion, dtype: float64
```

Классы сбалансированы, разница между ними составляет ~6–7% — это нормально и не требует дополнительных мер (например, балансировки или взвешивания классов).

Для оптимизации параметров моделей определено пространство гиперпараметров

Модель	Гиперпараметры
RandomForestClassifier	n_estimators: (100, 300) max_depth: (3, 15) min_samples_split: (2, 10) min_samples_leaf: (1, 4) max_features: [sqrt, log2] criterion: [gini, entropy]
HistGradientBoostingClassifier	learning_rate: (0.01, 0.3) max_iter: (50, 300) max_depth: (3, 15) l2_regularization: (0.1, 10.0) min_samples_leaf: (1, 20)
XGBClassifier	learning_rate: (0.01, 0.3) n_estimators: (100, 300) max_depth: (3, 12) min_child_weight: (1, 10) subsample: (0.6, 1.0) colsample_bytree: (0.6, 1.0) gamma: (0, 5) reg_alpha: (0.1, 10) reg_lambda: (0.1, 10)
LGBMClassifier	learning_rate: (0.01, 0.3) n_estimators: (100, 300) num_leaves: (20, 200) max_depth: (3, 15) min_child_samples: (5, 100) subsample: (0.6, 1.0) colsample_bytree: (0.6, 1.0) reg_alpha: (0.1, 10) reg_lambda: (0.1, 10)
CatBoostClassifier	learning_rate: (0.01, 0.3) depth: (3, 10) n_estimators: (100, 300) l2_leaf_reg: (1, 10) min_data_in_leaf: (1, 20) border_count: (32, 255)

## Обучение

- Используется **StratifiedKFold** с  $cv=5 \rightarrow$  сохраняем пропорции классов
- Оценивается модель на **5 фолдах**
- Считаются **несколько метрик** (accuracy, precision, recall, f1, roc\_auc)
- Возвращаются **усреднённые значения по тестовым фолдам**

По результатам обучения:

Результаты оценки моделей:

	Model	Best_F1_Val	Accuracy	Precision	Recall	F1	ROC_AUC
1.	HistGradientBoostingClassifier	0.663661	0.68875	0.671054	0.658667	0.663661	0.686980
2.	CatBoostClassifier	0.660906	0.68750	0.671905	0.650667	0.660906	0.685333
3.	RandomForestClassifier	0.660779	0.69000	0.678245	0.645333	0.660779	0.687373
4.	LGBMClassifier	0.651094	0.67500	0.654683	0.648000	0.651094	0.673412
5.	XGBClassifier	0.649600	0.67500	0.657721	0.642667	0.649600	0.673098

### 1. Лучшая модель: HistGradientBoostingClassifier

F1 Score = 0.6637

ROC AUC = 0.6870

Лучший результат среди всех моделей.

Хорошо сбалансирована между precision и recall.

Модель из sklearn, простая в использовании, устойчива к переобучению.

### 2. CatBoostClassifier и RandomForestClassifier близки к лидеру

F1  $\sim 0.66 \rightarrow$  показывают почти такое же качество.

CatBoost более интерпретируем (например, через SHAP).

Random Forest устойчив к шуму и не требует тонкой настройки.

### 3. LGBM и XGBoost показывают немного худшие результаты

F1  $\sim 0.65$

Возможно, это связано с особенностями датасета:

слабая зависимость от деревьев с высоким весом,

хорошее качество данных без выбросов.

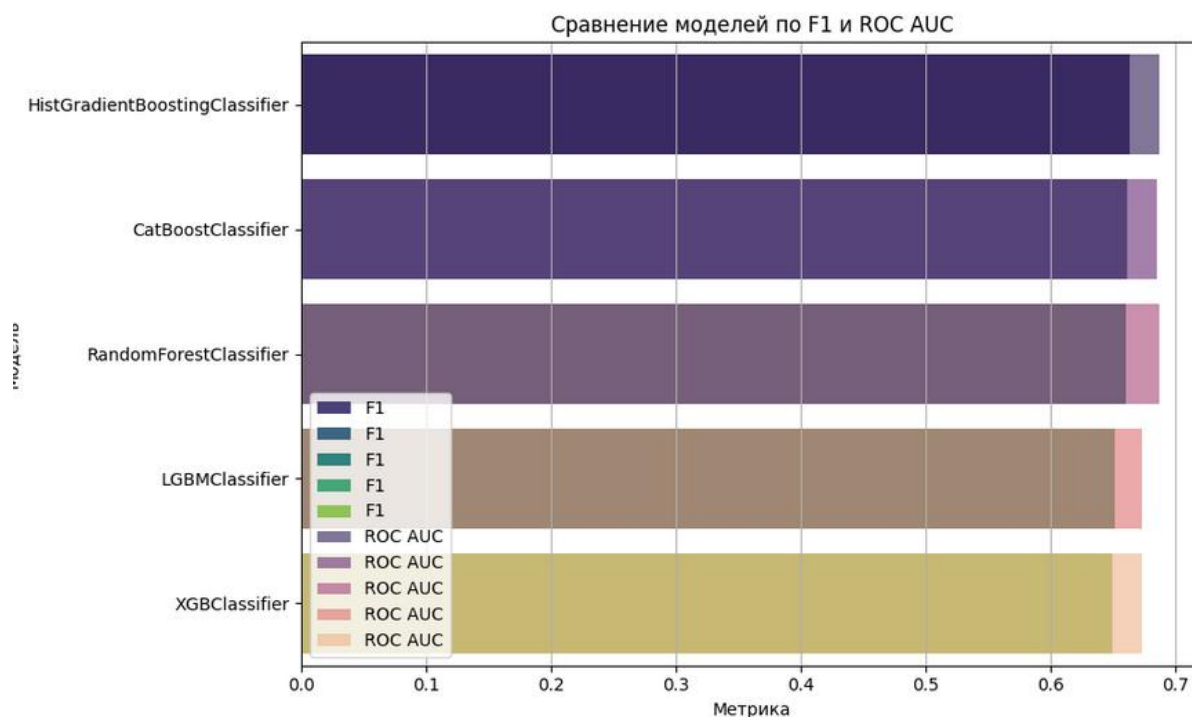
### 4. Модели сошлись за малое число trial'ов

Уже после  $n\_trials=10$  достигнут плато по метрике F1.

Это говорит о том, что данные имеют структуру, которую модели находят быстро.

### 5. Задача имеет потолок качества $\sim 0.66-0.67$ F1

Все модели группируются вокруг этого значения.



После оптимизации гиперпараметров с помощью Optuna, лучшие параметры были получены для каждой модели. Однако, эти параметры были найдены на основе кросс-валидации. Для получения финальных оценок качества моделей на независимой тестовой выборке, было решено:

Обучить каждую модель заново на всей обучающей выборке ( $X_{train}$ ,  $y_{train}$ ) с использованием лучших параметров.  
Оценить качество моделей на независимой тестовой выборке ( $X_{test}$ ,  $y_{test}$ ).

Это позволило получить более точные и стабильные оценки качества моделей, так как:

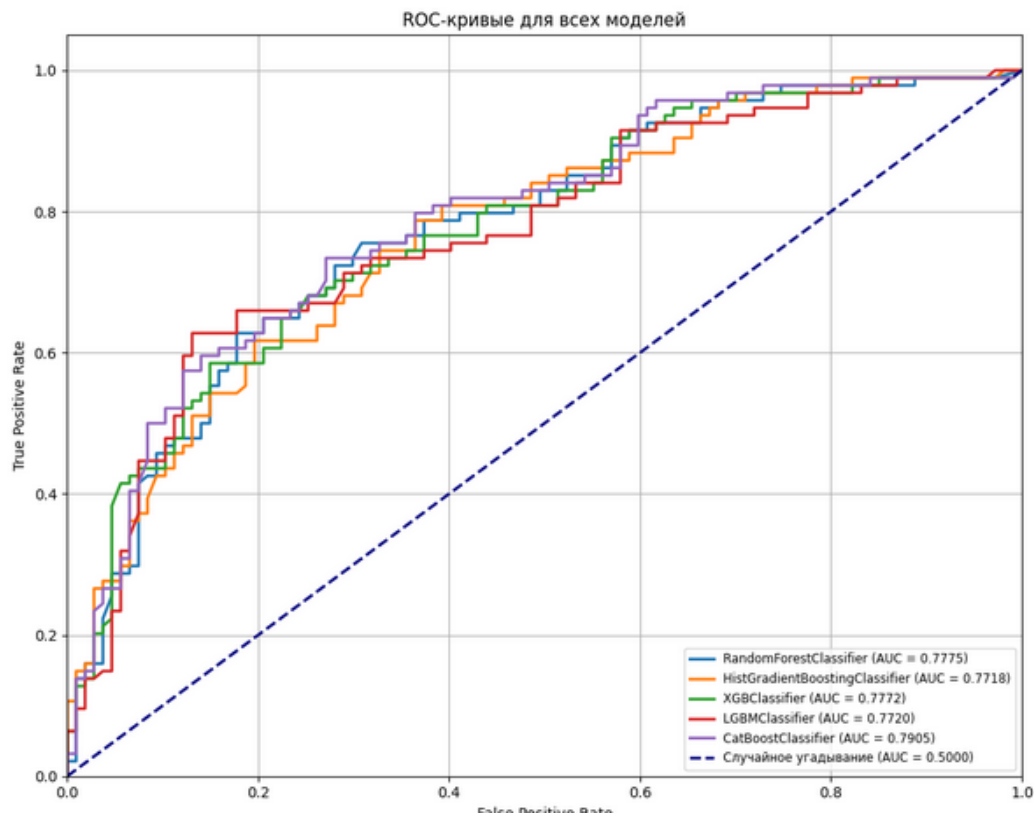
Модели обучались на полной обучающей выборке, что обычно приводит к лучшему качеству предсказаний.

Качество моделей было проверено на независимых данных, что гарантирует отсутствие переобучения

### Сравнение метрик до и после повторного обучения

#### Сравнение метрик до и после повторного обучения

Модель	F1 (валидация)	F1 (тест)	ROC AUC (валидация)	ROC AUC (тест)
HistGradientBoostingClassifier	0.6637	0.7718	0.6870	0.7718
CatBoostClassifier	0.6609	0.7953	0.6853	0.7953
RandomForestClassifier	0.6608	0.7775	0.6874	0.7775
LGBMClassifier	0.6511	0.7720	0.6734	0.7720
XGBClassifier	0.6496	0.7720	0.6731	0.7720



После повторного обучения на всей обучающей выборке CatBoostClassifier продемонстрировал лучший результат по обоим метрикам:  $F1 = 0.7953$  и  $ROC AUC = 0.7953$ . HistGradientBoostingClassifier также показал высокое качество ( $F1 = 0.7718$ ,  $ROC AUC = 0.7718$ ), но немного уступает CatBoost. Остальные модели (RandomForest, LGBM, XGBoost) имеют близкие значения метрик ( $\sim 0.77$ ).

### Возможные причины снижения метрик при кросс-валидации

- 1. Маленький объём данных на каждом фолде**  
При кросс-валидации модель обучается не на всех данных, а только на части. Это может привести к недообучению и занижению метрик.
- 2. Высокая вариативность между фолдами**  
Некоторые фолды могут быть сложными для модели (например, содержать редкие или шумные примеры), что ведёт к нестабильным результатам.
- 3. Слишком простая модель**  
Модель может не успевать "захватить" сложные зависимости на отдельных фолдах, особенно если данные требуют глубокого анализа.
- 4. Низкая информативность признаков**  
Если признаки слабо связаны с целевой переменной, это проявляется в низких метриках при кросс-валидации.
- 5. Переобучение на конкретные фолды**  
Модель может хорошо работать на одном фолде, но плохо — на другом. Это говорит о нестабильности и необходимости регуляризации.
- 6. Выбор порога классификации по умолчанию**  
При кросс-валидации используется жёсткий порог 0.5, который может быть не оптимальным для F1 Score

Таким образом, в качестве рекомендации можно предложить использовать модели обученные на всем train – наборе.

## Глава 6

### Классификация IC50> медианы

При подборе оптимальной модели для поиска соединений для которых IC5 превышает медианное значение выборки дополнительно мы изучим применение ансамбля из моделей.

Мы будем работать со следующим набором моделей

Модель	Библиотека	Классификатор / Регрессор	Основные преимущества
Random Forest	sklearn.ensemble	RandomForestClassifier	Устойчивость к переобучению, интерпретируемость, хорошее качество без тюнинга
HistGradientBoosting	sklearn.ensemble	HistGradientBoostingClassifier	Поддержка NaN, быстрое обучение, стабильность
XGBoost	xgboost	XGBClassifier	Высокая точность, гибкий тюнинг гиперпараметров
LightGBM (LGBM)	lightgbm	LGBMClassifier	Наивысшая скорость, эффективное дерево-рост
CatBoost	catboost	CatBoostClassifier	Отличная стабильность, автоматическая обработка числовых признаков

Метрика	Описание
---------	----------

Для оценки качества ансамблевых моделей будут использоваться те же метрики, что и на этапе оценки базовых моделей: Accuracy, Precision, Recall, F1 Score и ROC AUC , чтобы обеспечить сопоставимость результатов

Метрика	Описание
Accuracy	Доля правильных предсказаний среди общего числа
Precision (Точность)	Доля верно предсказанных положительных объектов среди всех предсказанных положительных
Recall (Полнота)	Доля верно предсказанных положительных объектов среди всех реальных положительных
F1 Score	Среднее гармоническое между Precision и Recall
ROC AUC	Площадь под ROC-кривой; отражает способность модели отличать классы при разных порогах

#### Построение ансамблевых моделей

После завершения этапа сравнительного анализа индивидуальных моделей машинного обучения следующим логичным шагом является построение ансамблевых решений . Ансамбли позволяют объединить предсказания нескольких базовых моделей с целью улучшения обобщающей способности, повышения устойчивости модели к шуму и снижения риска переобучения.

Существуют несколько подходов к построению ансамблей

**\*\*Усреднение вероятностей классов (Averaging\*\*)** — простой, но эффективный метод, при котором финальное предсказание получается как среднее значение вероятностей положительного класса от всех моделей.

**\*\*Голосование (Voting)\*\*** — комбинирование предсказаний классов с использованием жесткого (hard) или мягкого (soft) голосования.

**\*\*Стекинг (Stacking)\*\*** — более сложный подход, при котором предсказания базовых моделей используются как признаки для мета-модели, которая обучается предсказывать целевую переменную.

**\*\*\*Взвешенное голосование / усреднение\*\*** — использование весов для моделей на основе их качества на валидационной выборке.

Мы использовали два типа ансамблей: **Soft Voting** и **Stacking**. При этом не ограничились простым усреднением вероятностей, а применили более сложные и устойчивые методы комбинирования моделей

Каждый из этих подходов имеет свои преимущества и ограничения, а их сравнение позволит выбрать наиболее эффективное решение для задачи прогнозирования токсичности химических соединений по значению CC50.

Результаты:

Общая сводная таблица метрик моделей и ансамблей/n				
	Группа	Модель	F1	ROC AUC
0	Базовые	Random Forest	0.7418	0.8448
1	Базовые	HistGradientBoosting	0.7453	0.8529
2	Базовые	XGBoost	0.7324	0.8440
3	Базовые	LightGBM	0.7393	0.8478
4	Базовые	CatBoost	0.7349	0.8500
5	Ансамбли моделей	Soft Voting	0.7238	0.8550
6	Ансамбли моделей	Stacking	0.7379	0.8548

Изучение таблицы с результатами моделей и анализ ROC-кривых позволяют по-разному увидеть результаты. Выводы на основе табличного анализа:

- **HistGradientBoosting** показала лучший F1 Score, что говорит о хорошем балансе между точностью и полнотой.
- **LightGBM** имеет чуть меньший F1 Score, но самый высокий ROC AUC среди базовых моделей, что означает, что она хорошо ранжирует объекты.
- Все модели продемонстрировали сопоставимые результаты, без явного лидера.

#### ROC AUC и F1 Score

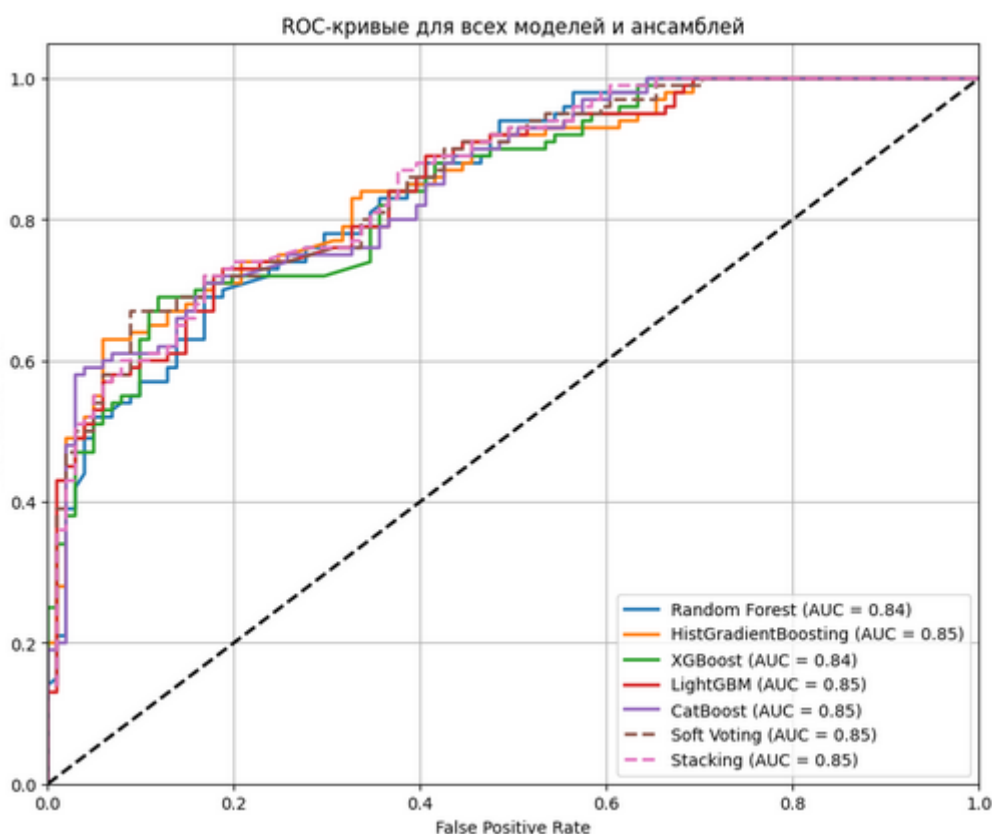
- ROC AUC у ансамблей выше, чем у большинства базовых моделей, что свидетельствует о том, что они лучше ранжируют классы.
- Однако F1 Score ансамблей ниже, чем у лучших индивидуальных моделей, что указывает на их невыигрышность в задаче баланса ошибок.

#### Итоговое заключение

На текущих данных ансамблирование не дало существенного улучшения качества по сравнению с лучшей индивидуальной моделью — **HistGradientBoosting**. Тем не менее:

- Ансамбли показали хороший ROC AUC, что говорит об их способности правильно ранжировать примеры. Это может быть полезно при дальнейшей настройке порога или в задачах, где важна оценка уверенности, а не жёсткое разделение на классы.
- Если цель — максимизация F1 Score, то достаточно использовать **HistGradientBoosting** как самостоятельную модель.

### Анализ ROC- кривых



### Вывод по анализу ROC-кривых и метрик моделей

Анализ ROC-кривой даёт новую пищу для размышлений. Несмотря на численно высокие метрики, **HistGradientBoosting** имеет сильный провал в центре кривой, что указывает на нестабильность модели в области средних вероятностей. Это особенно важно, если мы ориентируемся на F1 Score, поскольку именно в этой зоне формируется баланс между Precision и Recall.

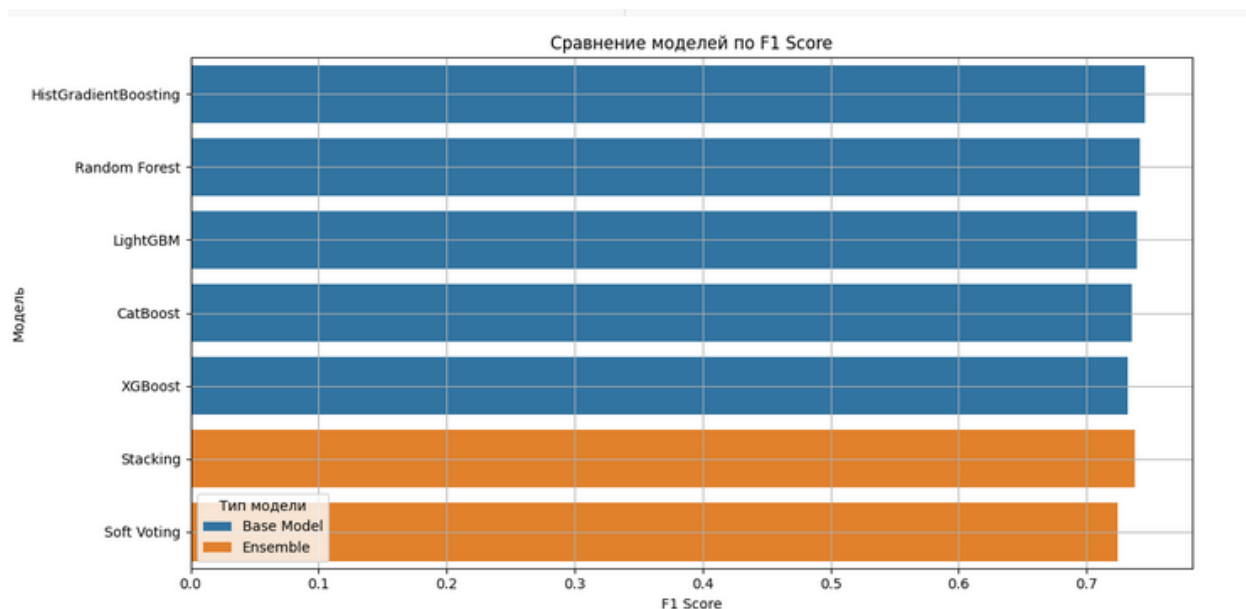
Такое поведение может быть связано с тем, что модель:

- Слишком уверенно классифицирует объекты в крайних случаях,
- Игнорирует сложные или неоднозначные примеры,
- Может быть чувствительной к шуму или переобучаться на определённых участках пространства признаков.



В свою очередь, ансамбли (Soft Voting и Stacking) показывают более гладкие и стабильные кривые, без резких провалов. Это говорит о том, что они:

- Лучше улавливают общий паттерн,
- Равномернее распределяют уверенность по всему диапазону,
- Могут давать более надёжные предсказания в задачах, где важен баланс между полнотой и точностью.



Таким образом, хотя **HistGradientBoosting** демонстрирует хороший результат по метрике ROC AUC, его поведение в центральной части кривой вызывает вопросы. Ансамбли, напротив, обеспечивают более равномерное качество и могут быть предпочтительнее, особенно если цель — максимизация F1 Score и стабильность предсказаний.

### Итоговое заключение

Хотя **HistGradientBoosting** показала наивысший F1 Score, её поведение на ROC-кривой выявило провал в центральной области, что может говорить о нестабильности модели при классификации объектов со средней уверенностью. Это особенно критично, если важен баланс между точностью и полнотой.

В свою очередь, стекинг обеспечивает более равномерное качество по всему диапазону вероятностей и демонстрирует высокий уровень ROC AUC. Он позволяет:

- Комбинировать силы нескольких моделей,
- Снижать дисперсию предсказаний,
- Получать более обобщаемое и устойчивое решение.

Таким образом, для задач прогнозирования токсичности химических соединений стекинг является предпочтительным вариантом, обеспечивая надежные предсказания и стабильность в работе модели.

## Глава 7.

### Классификация SI выше медианы.

Для исследования мы отберем только модели показавшие лучшие результаты на прошлых этапах исследования

Модель	Библиотека	Классификатор / Регрессор	Основные преимущества
HistGradientBoosting	sklearn.ensemble	HistGradientBoostingClassifier	Поддержка NaN, быстрое обучение, стабильность
XGBoost	xgboost	XGBClassifier	Высокая точность, гибкий тюнинг гиперпараметров
CatBoost	catboost	CatBoostClassifier	Отличная стабильность, автоматическая обработка числовых признаков

Мы будем использовать расширенный пакт метрик

Метрика	Описание	Зачем нужна
Accuracy	Доля правильных предсказаний среди общего числа	Общая мера эффективности модели; удобна при сбалансированных классах. Может быть неинформативной при дисбалансе.
Precision	Доля верно предсказанных положительных объектов среди всех предсказанных положительных	Важна, когда ложные срабатывания критичны (например, дорого проверять вещества в реальности).
Recall	Доля верно предсказанных положительных объектов среди всех реальных положительных	Критична, если важно найти как можно больше истинных случаев (например, пропустить активное вещество нельзя).
F1 Score	Среднее гармоническое между Precision и Recall	Хорошая обобщающая метрика, особенно при дисбалансе классов. Балансирует между ошибками I и II рода.
ROC AUC	Площадь под ROC-кривой; отражает способность модели отличать классы при разных порогах	Показывает, насколько модель уверенно ранжирует объекты; полезна для сравнения моделей. Не зависит от порога классификации.
PR AUC	Площадь под Precision-Recall кривой	Альтернатива ROC AUC, чувствительнее к дисбалансу классов, показывает надежность предсказаний.
Balanced Accuracy	Усредненная точность между классами	Учитывает дисбаланс классов, более надежная альтернатива Accuracy.
Log Loss	Логарифмическая функция потерь	Оценивает качество вероятностных предсказаний; полезна при тонкой настройке моделей.
SHAP Importance	Вклад каждого признака в предсказание	Помогает интерпретировать, какие признаки оказывают наибольшее влияние на предсказание.

### Результаты полученные базовыми моделями

```
warnings.warn(msg, UserWarning)
```

	Accuracy	Precision	Recall	F1 Score	ROC AUC	\
HistGradientBoosting	0.696517	0.734940	0.61	0.666667	0.748267	
XGBoost	0.641791	0.655556	0.59	0.621053	0.708960	
CatBoost	0.661692	0.695122	0.57	0.626374	0.728168	

	Log Loss
HistGradientBoosting	0.709253
XGBoost	0.864174
CatBoost	0.609293

### Анализ метрик

**CatBoost** показал лучшие результаты по Accuracy (66.17%) и Precision (69.51%), но Recall у него остается на уровне 57%. Это говорит о высокой избирательности модели, которая может пропускать сложные случаи.

**HistGradientBoosting** продемонстрировал самую высокую ROC AUC (74.83%), что говорит о его способности ранжировать объекты по вероятности. F1 Score также высок — 66.67%, что указывает на хороший баланс между полнотой и точностью.

**XGBoost** имеет менее сбалансированные показатели: Accuracy (64.18%), Precision (65.56%), Recall (59%), ROC AUC (70.90%). Модель демонстрирует среднюю эффективность и более высокий Log Loss, что может говорить о менее уверенных вероятностных оценках. e (68.38%), что может говорить о менее уверенном ранжировании объектов.

### Анализ ROC-кривых

#### 1. CatBoost

Кривая : CatBoost демонстрирует более плавную кривую , что указывает на её способность уверенно ранжировать объекты по вероятности. Стабильность : В отличие от других

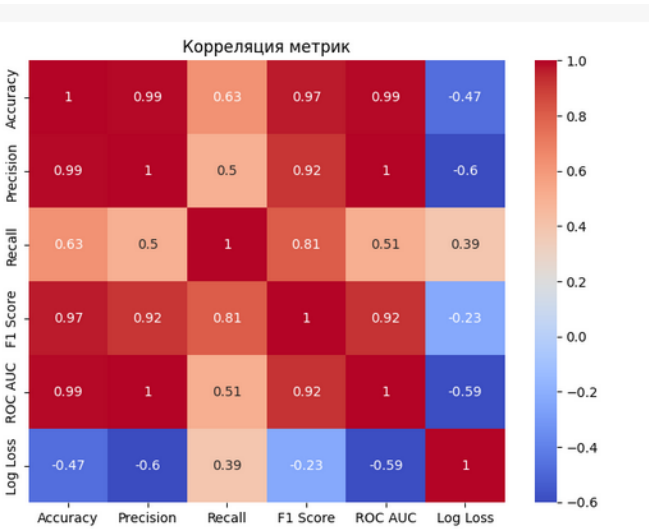
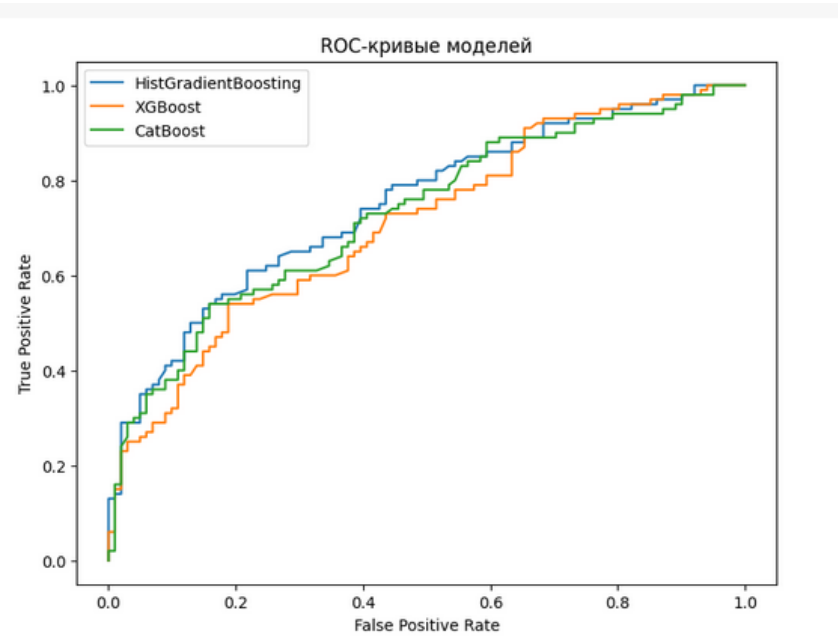
моделей, CatBoost показывает меньше колебаний в области средних значений False Positive Rate (FPR), что говорит о её высокой устойчивости при различных порогах классификации.

### 2. XGBoost

Кривая : XGBoost имеет схожую кривую с CatBoost, но её форма немного менее плавная.  
Недостатки : В центральной части кривой видны небольшие "ступеньки", что может указывать на некоторую нестабильность в ранжировании объектов с средними вероятностями .

### 3. HistGradientBoosting

Кривая : HistGradientBoosting демонстрирует наиболее плавную кривую среди всех моделей, особенно в области высоких True Positive Rates (TPR). Провал в центральной части : Однако в области средних вероятностей ( $FPR \approx 0.4-0.6$ ) виден легкий провал , что может говорить о несколько меньшей стабильности в этой зоне по сравнению с CatBoost.



## Анализ матрицы корреляции метрик

1. Высокая корреляция между Precision и Accuracy Эти метрики сильно связаны, что означает, что точность предсказаний зависит от правильного определения положительного класса. Однако высокий Precision при низком Recall может свидетельствовать о том, что модель склонна избегать ложных срабатываний, пропуская значимые случаи.
2. Низкая корреляция между Recall и ROC AUC Несмотря на важность ROC AUC для ранжирования, слабая связь с Recall говорит о том, что модели, вероятно, испытывают сложности с балансом между полнотой и уверенностью предсказаний. Возможно, стоит оптимизировать порог классификации или использовать дополнительные методы балансировки.
3. Обратная связь между Log Loss и Precision Если Log Loss велик, это может означать, что модель недостаточно уверена в своих предсказаниях. Высокая точность предсказаний приводит к низкому значению Log Loss, но если модель слишком агрессивна в классификации, она может пропускать сложные случаи.
4. Balanced Accuracy имеет слабую связь с другими метриками Balanced Accuracy учитывает дисбаланс классов и может давать дополнительную информацию о модели. Это говорит о том, что данную метрику стоит учитывать при оптимизации параметров алгоритмов, особенно если классы распределены неравномерно.
5. Precision-Recall AUC как более точная альтернатива ROC AUC Эта метрика может быть полезной при анализе моделей, так как она лучше отражает способность алгоритмов работать с дисбалансом классов. Ее можно включить в подбор оптимальных параметров.

## Общий вывод:

**CatBoost** — наиболее устойчивая модель по ROC-кривой: её кривая выглядит плавной и без резких скачков. **HistGradientBoosting** — демонстрирует лучшее ранжирование в целом, но имеет небольшую нестабильность в области средних вероятностей. **XGBoost** — имеет среднюю производительность, но требует более тщательной настройки гиперпараметров для улучшения стабильности.

На следующем этапе был произведен подбор параметров моделей с оптимизацией Optuna

Результаты Optuna-оптимизации:

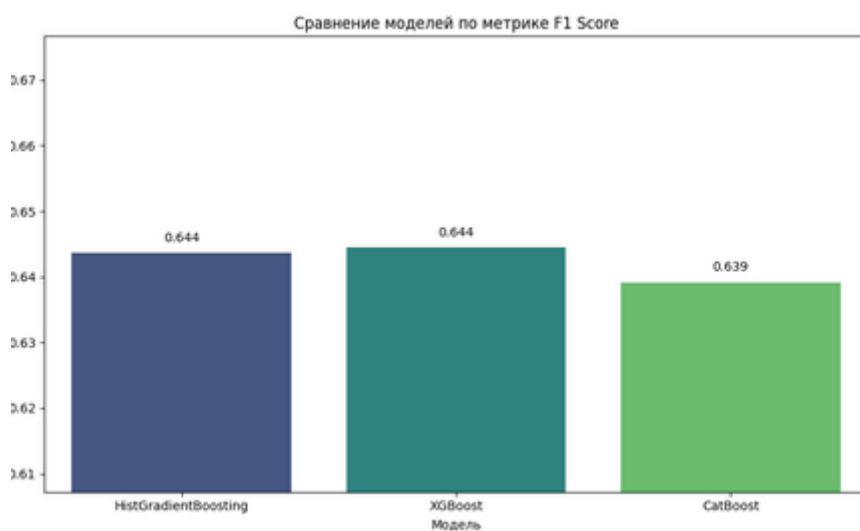
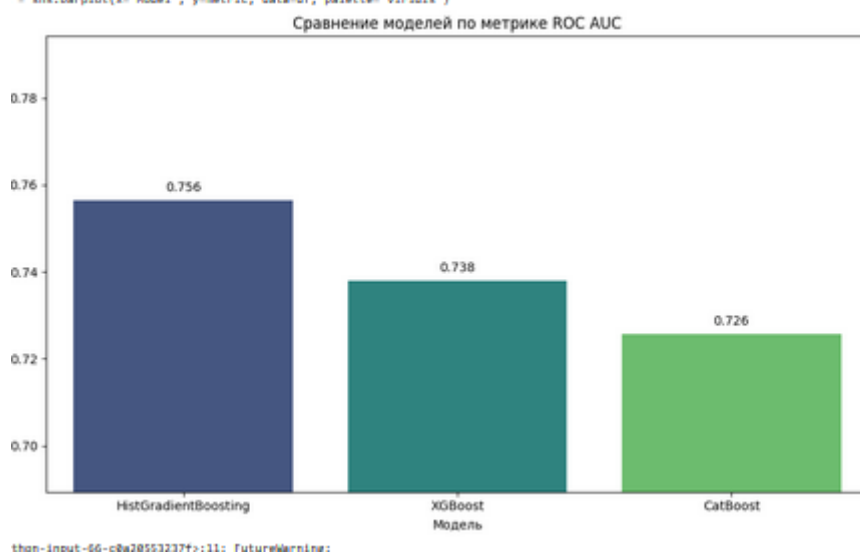
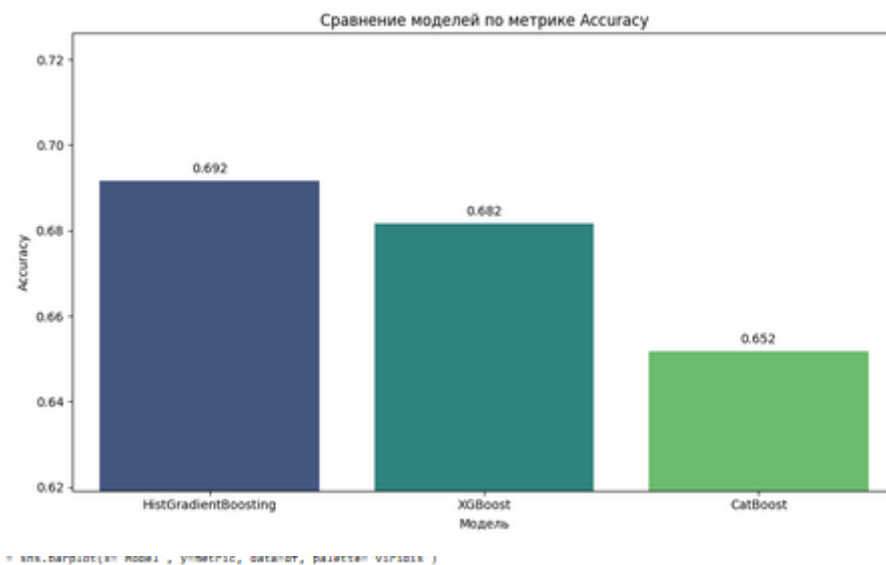
	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	HistGradientBoosting	0.691542	0.756757	0.56	0.643678	0.756386
1	XGBoost	0.681592	0.725000	0.58	0.644444	0.737871
2	CatBoost	0.651741	0.659574	0.62	0.639175	0.725693

Log Loss

0	0.607716
1	0.607983
2	0.724848

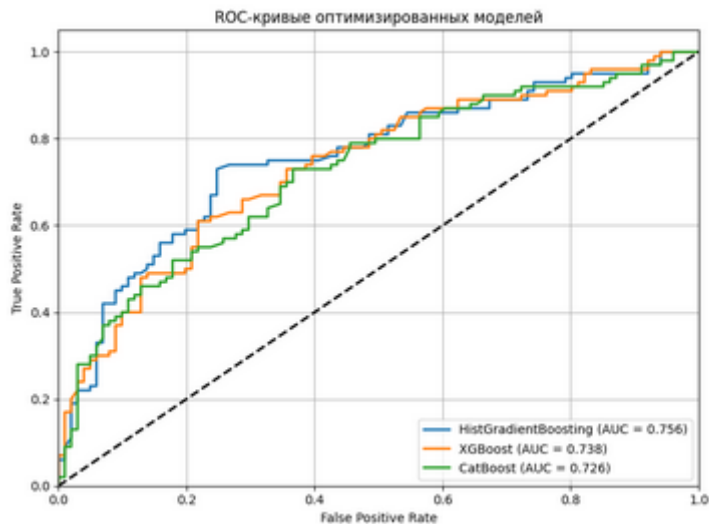
## Проведем сравнение моделей по наиболее значимым метрикам

Построим графики для ROC AUC, F1score.



Лидером оказывается **HistGradientBoosting**, который после оптимизации обгоняет все другие модели, включая прежнего лидера **CatBoost**

Визуально оценим ROC- кривые



### Ключевые выводы из графика:

#### HistGradientBoosting :

Лучшая кривая среди всех моделей

Наибольший AUC: 0.756

Стабильно выше случайной модели (диагональная линия)

#### XGBoost :

Вторая по качеству кривая

AUC: 0.738

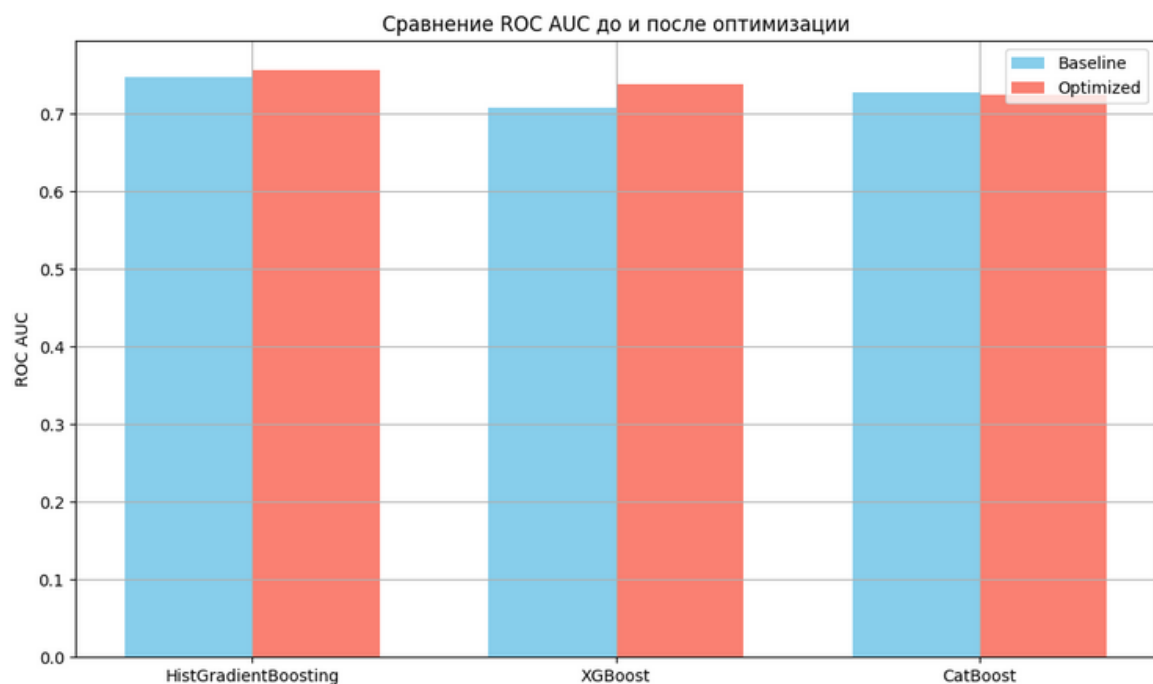
Немного уступает HistGradientBoosting, но всё ещё хорошо ранжирует объекты

#### CatBoost :

Третья по качеству кривая

AUC: 0.726

Устойчиво работает, но немного отстаёт от других моделей



Metric	HistGradientBoosting	XGBoost	CatBoost
Accuracy_Baseline	0.6965	0.6418	0.6617
Precision_Baseline	0.7349	0.6556	0.6951
Recall_Baseline	0.6100	0.5900	0.5700
F1 Score_Baseline	0.6667	0.6211	0.6264
ROC AUC_Baseline	0.7483	0.7090	0.7282
Log Loss_Baseline	0.7093	0.8642	0.6093
Accuracy_Optimized	0.6915	0.6816	0.6517
Precision_Optimized	0.7568	0.7250	0.6596
Recall_Optimized	0.5600	0.5800	0.6200
F1 Score_Optimized	0.6437	0.6444	0.6392
ROC AUC_Optimized	0.7564	0.7379	0.7257
Log Loss_Optimized	0.6077	0.6080	0.7248
Accuracy_Diff_abs	-0.0050	0.0398	-0.0100
Accuracy_Diff_pct	-0.7143	6.2016	-1.5038
Precision_Diff_abs	0.0218	0.0694	-0.0355
Precision_Diff_pct	2.9685	10.5932	-5.1138
Recall_Diff_abs	-0.0500	-0.0100	0.0500
Recall_Diff_pct	-8.1967	-1.6949	8.7719
F1 Score_Diff_abs	-0.0230	0.0234	0.0128
F1 Score_Diff_pct	-3.4483	3.7665	2.0438
ROC AUC_Diff_abs	0.0081	0.0289	-0.0025
ROC AUC_Diff_pct	1.0850	4.0779	-0.3399
Log Loss_Diff_abs	-0.1015	-0.2562	0.1156
Log Loss_Diff_pct	-14.3160	-29.6458	18.9655

## Сравнительный анализ базовых моделей и оптимизации

### HistGradientBoosting:

- Увеличилась уверенность модели: Log Loss ↓ на 14.3%
- Значительно вырос ROC AUC: ↑ на 1.1%
- Однако снизился Recall: ↓ на 8.2%, что может быть критичным, если важно находить больше истинных позитивов.

### XGBoost:

- Показал наибольший прогресс по всем метрикам:
  - Увеличение Accuracy на 6.2%
  - Precision на 10.6%
  - F1 Score на 3.7%
  - ROC AUC на 4.1%
- Особенно заметное улучшение Log Loss: ↓ на 29.6%

### CatBoost:

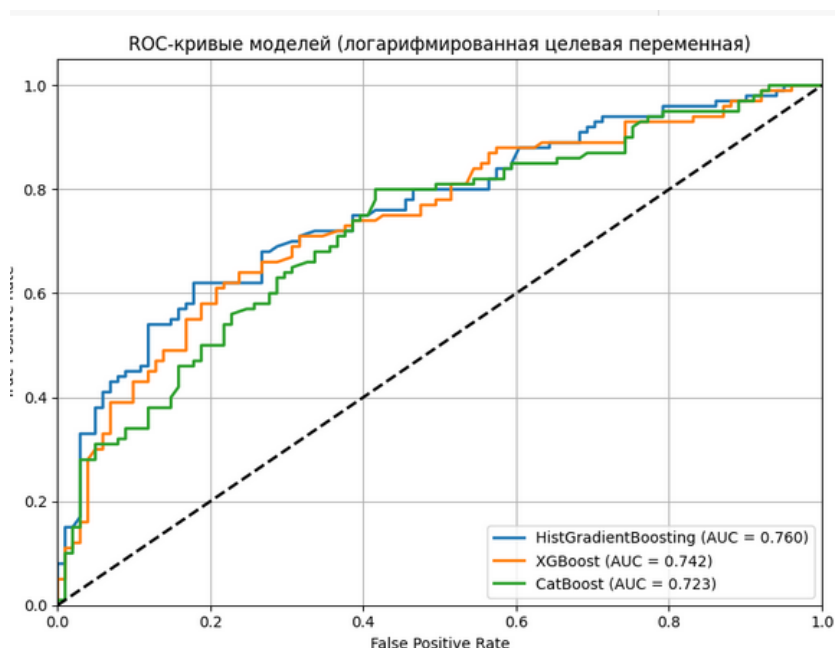
- Улучшил Recall: ↑ на 8.8%

- Повысил F1 Score: ↑ на 2.0%, что делает её хорошим выбором для задач, где важны находить больше истинных позитивов.
- Однако снизилась точность (Precision ↓ на 5.1%)
- И увеличился Log Loss: ↑ на 19.0%, что может быть недостатком.

**Лучшая модель после оптимизации:**  
**XGBoost** демонстрирует наиболее стабильный прирост по всем ключевым метрикам, особенно по Log Loss и ROC AUC.

**Проведем этап логарифмирования целевой переменной и повторный поиск оптимизации optuna**

Модель	Версия целевой	Accuracy	Precision	Recall	F1 Score	ROC AUC	Log Loss
HistGradientBoosting	y	0.6915	0.7568	0.56	0.6437	0.7564	0.6077
HistGradientBoosting	log(y)	0.7114	0.7561	0.62	0.6813	0.7600	0.5784
XGBoost	y	0.6816	0.7250	0.58	0.6444	0.7379	0.6080
XGBoost	log(y)	0.6965	0.7349	0.61	0.6667	0.7424	0.6043
CatBoost	y	0.6517	0.6596	0.62	0.6392	0.7257	0.7248
CatBoost	log(y)	0.6667	0.6813	0.62	0.6492	0.7229	0.7628



**Выводы по сравнению:**

### 1. HistGradientBoosting

ROC AUC ↑ на 0.48% , Log Loss ↓ на 4.81% Recall вырос на 10.71% , что сильно влияет на F1 Accuracy и F1 также выросли Вывод: Логарифмирование существенно улучшило модель , особенно по полноте и уверенности в предсказаниях.



## 2. XGBoost

Все метрики выросли, особенно F1 (+3.46%) и ROC AUC (+0.61%) Небольшое снижение Log Loss (на -0.61%) Вывод: Преобразование целевой переменной дало стабильный прирост по всем метрикам.

## 3. CatBoost

Accuracy, Precision, F1 выросли, но ROC AUC немного упал Log Loss увеличился на 6.2% — это может быть критично, если важна уверенность в вероятностях Вывод: CatBoost стал чуть точнее, но потерял стабильность в ранжировании объектов

Лучшая модель после логарифмирования: HistGradientBoosting

Наибольший прирост по ключевым метрикам

Уверенность в предсказаниях (Log Loss ↓)

Самый значительный рост Recall и F1 Score

### Применение нейросетей

Для дальнейших экспериментов с нейросетями выбрана логарифмированная целевая переменная ( $\log(y)$ ).

Почему:

Все рассмотренные модели (**HistGradientBoosting, XGBoost, CatBoost**) демонстрируют стабильное улучшение по ключевым метрикам, особенно по ROC AUC, F1 Score и Log Loss. Наибольший прирост показывает модель HistGradientBoosting — это указывает на то, что преобразование положительно влияет на предсказательную способность моделей. Логарифмирование позволило сделать распределение целевой переменной более равномерным, что может способствовать лучшей обучаемости как градиентного бустинга, так и нейросетевых моделей.

### Запуск с минимальными параметрами

На данном этапе был выполнен автоматический подбор модели для табличных данных с помощью фреймворка AutoKeras. В процессе было произведено не более 5 попыток (`max_trials=5`) для поиска лучшей архитектуры нейронной сети. После обучения лучшая найденная модель была сохранена для дальнейшего использования и оценки качества.

Мы используем одну и ту же функцию для расчёта метрик качества для всех моделей. Эта функция считает такие метрики, как точность (accuracy), полноту (recall), precision, F1-меру, ROC AUC и log loss.

Однако оказалось, что AutoKeras возвращает из метода `.predict()` не готовые классы (например, 0 или 1), а вероятности принадлежности к классам (например, значения от 0 до 1, вроде 0.85).

Это вызывает ошибку при использовании некоторых метрик, потому что они ожидают на вход именно классы, а не вероятности.

Чтобы всё работало корректно и результаты были сравнимы с другими моделями, мы бинаризуем предсказания AutoKeras по порогу 0.5. То есть, если значение больше или равно 0.5 — считаем это за класс 1, иначе — 0.

Таким образом, мы сохраняем единый формат входных данных для функции расчёта метрик и можем честно сравнивать все модели между собой.

В результате 5 попыток была выбрана модель

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 185)	0
cast (Cast)	(None, 185)	0
cast_to_float32 (CastToFloat32)	(None, 185)	0
dense (Dense)	(None, 32)	5,952
re_lu (ReLU)	(None, 32)	0
dense_1 (Dense)	(None, 32)	1,056
re_lu_1 (ReLU)	(None, 32)	0
dropout (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33
classification_head_1 (Activation)	(None, 1)	0

Total params: 7,041 (27.50 KB)  
Trainable params: 7,041 (27.50 KB)  
Non-trainable params: 0 (0.00 B)

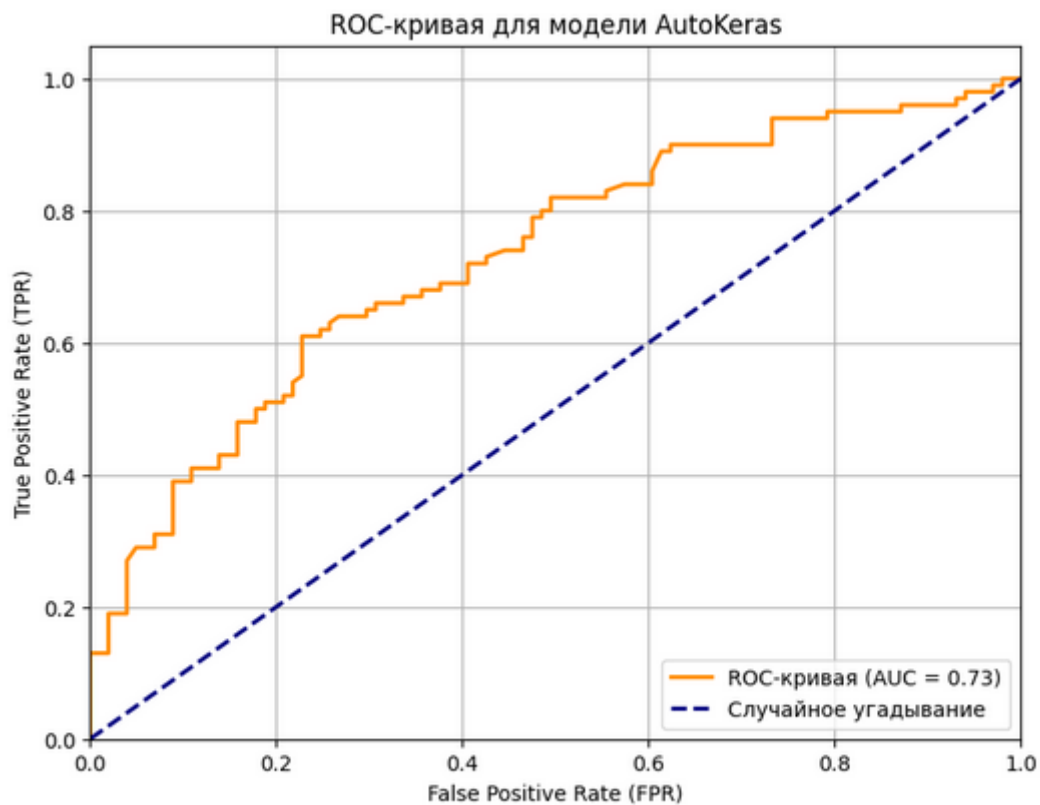
Была построена полносвязная нейронная сеть, состоящая из нескольких слоев с разным количеством нейронов. На каждом скрытом слое применялась функция активации ReLU для введения нелинейности. Для задачи бинарной классификации функция активации также использовалась на выходном слое.

В модели применялись операции преобразования типа данных (слои Cast), которые обеспечивают соответствие типа входных данных ожиданиям последующего слоя, гарантируя корректную работу всей сети.

Для предотвращения переобучения использовался слой Dropout — регуляризация, которая случайным образом отключает часть нейронов во время обучения. В данной модели Dropout действует непосредственно на выходе одного из скрытых слоев, уменьшая зависимость модели от конкретных нейронов и повышая её обобщающую способность.

```
Метрики модели AutoKeras:
      Accuracy Precision Recall F1 Score ROC AUC Log Loss
AutoKeras  0.691542   0.72619   0.61  0.663043  0.730149  0.63142
```

Сравнение метрик модели AutoKeras с предыдущими моделями показывает, что AutoKeras на данном этапе не продемонстрировал улучшения.



Модель	Accuracy	Precision	Recall	F1 Score	ROC AUC	Log Loss
<b>AutoKeras</b>	0.6915	0.7262	0.61	0.6630	0.7301	0.6314
HistGradientBoosting y	0.6915	0.7568	0.56	0.6437	0.7564	0.6077
HistGradientBoosting log(y)	0.7114	0.7561	0.62	0.6813	0.7600	0.5784
XGBoost y	0.6816	0.7250	0.58	0.6444	0.7379	0.6080
XGBoost log(y)	0.6965	0.7349	0.61	0.6667	0.7424	0.6043
CatBoost y	0.6517	0.6596	0.62	0.6392	0.7257	0.7248

#### Выводы:

- По точности (Accuracy) и F1 Score AutoKeras находится примерно на уровне лучших моделей, но не превосходит их.
- Метрики Precision у AutoKeras ниже, чем у HistGradientBoosting.
- Recall у AutoKeras лучше, чем у некоторых моделей, но ниже, чем у HistGradientBoosting с логарифмированием таргета.
- ROC AUC и Log Loss у AutoKeras хуже, чем у лучших моделей (HistGradientBoosting log(y) и XGBoost log(y)).
- Наилучшие результаты по большинству метрик показывает HistGradientBoosting с логарифмированием целевой переменной.

## Итог:

На этом этапе AutoKeras не смог превзойти предыдущие модели по ключевым метрикам. Для улучшения результатов решено рассмотреть оптимизацию гиперпараметров

## Расширение пространства параметров AutoKeras

Подбор более эффективной архитектуры нейросети

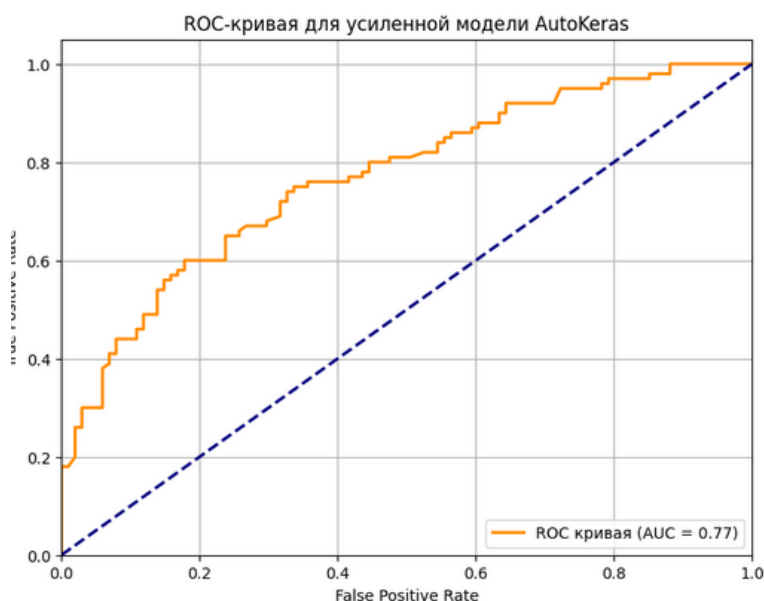
Для подбора более эффективной архитектуры нейросети, предназначенной для классификации веществ по признаку SI выше медианы, использовался фреймворк AutoKeras с блоком DenseBlock, обеспечивающим автоматический подбор структуры сети.

Общее количество попыток поиска (trials) было увеличено до 50, что позволило расширить пространство поиска и повысить вероятность нахождения оптимальной архитектуры. Также были добавлены:

Регуляризация модели (Dropout, L2 — через автоматический подбор)

Ранняя остановка (EarlyStopping) для предотвращения переобучения и ускорения поиска

Фреймворк поиска моделей остался прежним — поиск осуществлялся на основе DenseBlock + ClassificationHead, без ручного указания гиперпараметров в самом блоке, чтобы избежать ошибок совместимости в AutoKeras 2.0.0.



## Сравнение моделей по метрикам (log(y))

Модель	Accuracy	Precision	Recall	F1 Score	ROC AUC	Log Loss
HistGradientBoosting	0.7114	0.7561	0.62	0.6813	0.7600	0.5784
XGBoost	0.6965	0.7349	0.61	0.6667	0.7424	0.6043
CatBoost	0.6667	0.6813	0.62	0.6492	0.7229	0.7628
AutoKeras (Dense)	0.6915	0.7262	0.61	0.6630	0.7301	0.6314
AutoKeras_Enhanced	0.6965	0.7407	0.60	0.6630	0.7692	0.5643

Цель эксперимента: Определить наиболее эффективную модель для бинарной классификации, предсказывая, находится ли показатель SI (Solubility Index) выше медианы.

Использованные модели:

- Традиционные модели градиентного бустинга: HistGradientBoosting, XGBoost, CatBoost.
- Нейросеть: AutoKeras с автоматическим подбором архитектуры.
- Оптимизация гиперпараметров: Использование Optuna.
- Преобразование целевой переменной: Логарифмирование для улучшения стабильности и качества предсказаний.

Результаты:

- HistGradientBoostingClassifier:
  - ROC AUC: 0.7600
  - F1 Score: 0.6813
  - Log Loss: 0.5784
- AutoKeras\_Enhanced:
  - ROC AUC: 0.7692 (лучший результат)
  - F1 Score: 0.6630
  - Log Loss: 0.5643 (самая уверенная модель)

Вывод: Нейросеть, обученная с помощью AutoKeras, показала лучшие метрики по ROC AUC и Log Loss по сравнению с традиционными моделями градиентного бустинга. При этом время обучения составило всего 15-20 минут, что делает эту модель особенно привлекательной для дальнейших исследований.

Итог: Нейросеть AutoKeras\_Enhanced продемонстрировала сравнимые или даже лучшие результаты по сравнению с HistGradientBoosting, что подтверждает ее эффективность для данной задачи. Это ключевая находка вашего этапа, показывающая, что расширенный подход с использованием AutoKeras может быть предпочтительным для задач бинарной классификации.

## Глава 8

### Классификация SI более 8

На предыдущих этапах исследования мы отказались от ряда моделей, таких как SVM, Logistic Regression, из-за их относительной слабости и низкой эффективности при настройке с помощью подбора гиперпараметров. Хотя некоторые из этих моделей показывали сравнимые результаты «из коробки», мы сосредоточились на бустинговых моделях, поддерживающих гибкую настройку гиперпараметров с помощью Optuna и возможность автоматического сокращения количества признаков в процессе обучения.

Модель	Краткое описание
Random Forest Regressor	Ансамблевый метод на основе множества деревьев решений. Устойчив к переобучению, не требует тонкой настройки.

Модель	Краткое описание
Gradient Boosting Regressor	Последовательный ансамблевый метод, учитывающий ошибки предыдущих. Хорошая предсказательная способность, чувствителен к гиперпараметрам.
HistGradientBoostingRegressor	Усовершенствованная версия градиентного бустинга в scikit-learn. Быстро обучается, поддерживает пропуски, эффективен на больших данных.
XGBoost Regressor	Высокоэффективный градиентный бустинг с оптимизацией скорости и качества. Поддерживает регуляризацию и параллельные вычисления.
LGBMRegressor (LightGBM)	Быстрый и экономный по памяти градиентный бустинг от Microsoft. Эффективен при большом числе признаков.
CatBoostRegressor	Градиентный бустинг от Яндекса. Отлично работает с числовыми и категориальными данными, встроенная защита от переобучения.

Метрика	Описание
Accuracy	Доля правильных предсказаний среди общего числа
Precision (Точность)	Доля верно предсказанных положительных объектов среди всех предсказанных положительных
Recall (Полнота)	Доля верно предсказанных положительных объектов среди всех реальных положительных
F1 Score	Среднее гармоническое между Precision и Recall
ROC AUC	Площадь под ROC-кривой; отражает способность модели отличать классы при разных порогах

## Результаты запуска базовых моделей

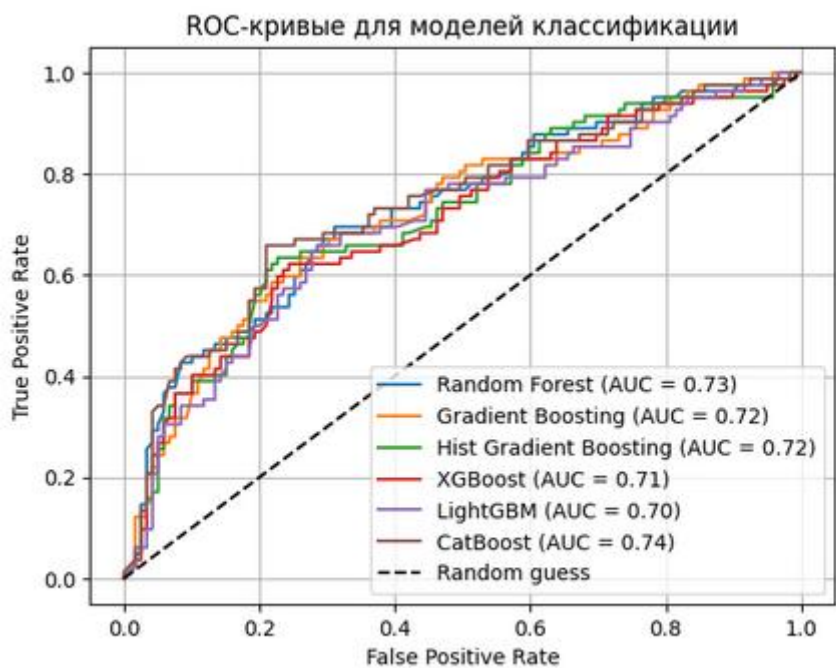
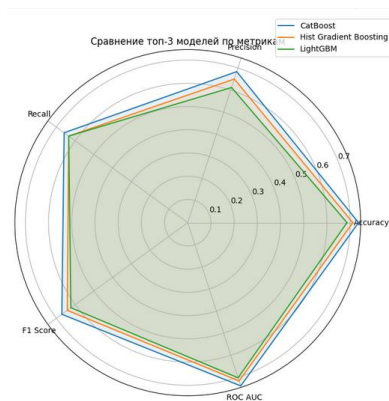
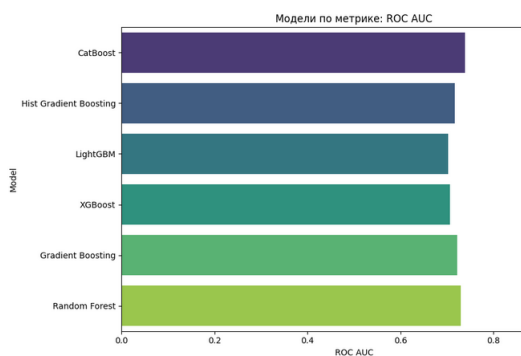
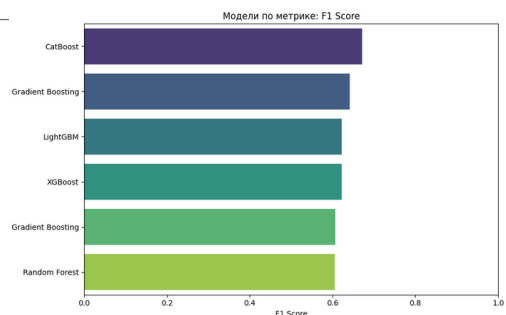
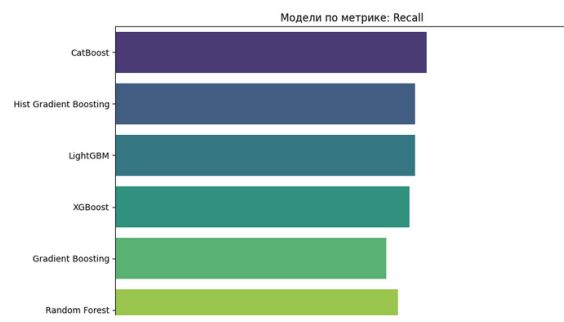
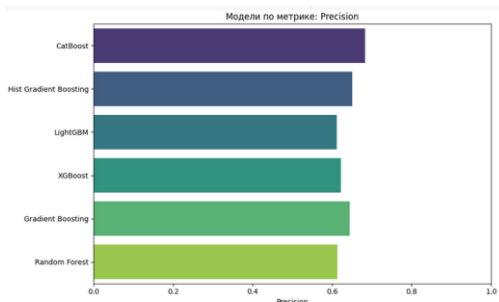
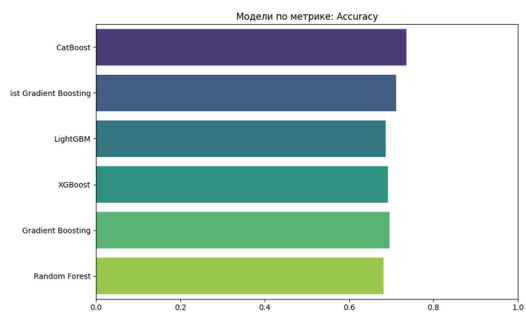
	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
5	CatBoost	0.736318	0.683544	0.658537	0.670807	0.738830
2	Hist Gradient Boosting	0.711443	0.650000	0.634146	0.641975	0.716694
4	LightGBM	0.686567	0.611765	0.634146	0.622754	0.702142
3	XGBoost	0.691542	0.621951	0.621951	0.621951	0.706241
1	Gradient Boosting	0.696517	0.643836	0.573171	0.606452	0.721306
0	Random Forest	0.681592	0.612500	0.597561	0.604938	0.729453

## Выводы

**CatBoost** — лидер по совокупности метрик: самый высокий F1 и ROC AUC.

HistGradientBoostingClassifier и XGBoost — стабильные модели с хорошим балансом между точностью и полнотой.

Random Forest и Gradient Boosting показывают худший результат, хотя и остаются приемлемыми для начального анализа

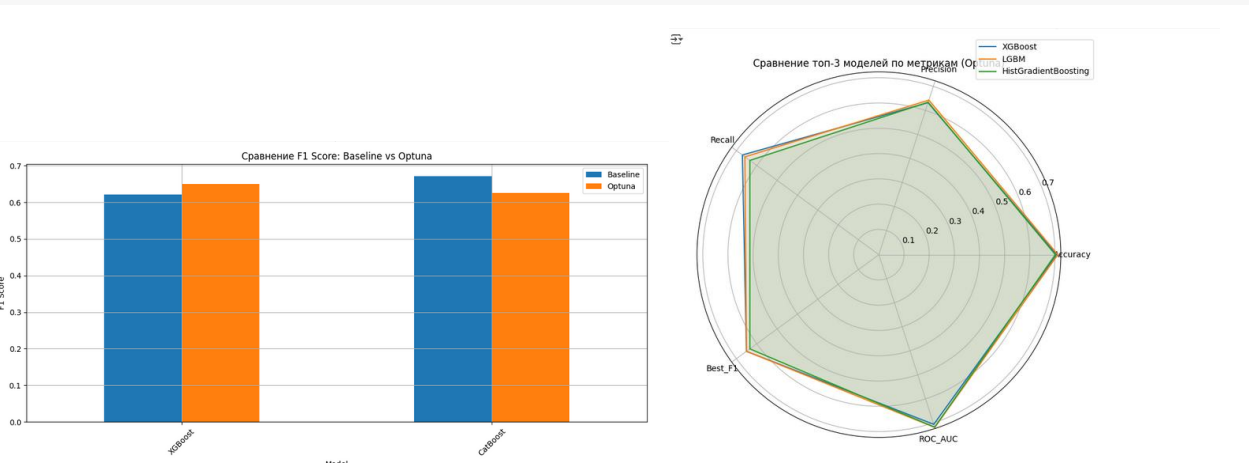


После предварительного анализа и сравнения нескольких бустинговых моделей было установлено, что CatBoost демонстрирует наилучшие результаты по метрикам F1 Score и ROC AUC

Оптимизация подбора параметров optuna с регуляризацией не позволил существенно улучшить метрики

→

Результаты Optuna:							
	Model	Best_Params	Best_F1	Accuracy	Precision	Recall	ROC_AUC
0	XGBoost	{'learning_rate': 0.1, 'max_depth': 7, 'n_esti...	0.650888	0.706468	0.632184	0.670732	0.706036
1	LGBM	{'learning_rate': 0.2, 'num_leaves': 127, 'max...	0.650602	0.711443	0.642857	0.658537	0.718846
2	HistGradientBoosting	{'learning_rate': 0.2, 'max_depth': None, 'l2_...	0.634146	0.701493	0.634146	0.634146	0.717719
3	CatBoost	{'learning_rate': 0.2, 'depth': 5, 'n_estimato...	0.626506	0.691542	0.619048	0.634146	0.699477
4	GradientBoosting	{'learning_rate': 0.2, 'n_estimators': 200, 'm...	0.625767	0.696517	0.629630	0.621951	0.723253
5	RandomForest	{'n_estimators': 100, 'max_depth': None, 'min_...	0.604938	0.681592	0.612500	0.597561	0.729453



Дополнительно для тех же моделей проведена оптимизация в большем пространстве параметров.

Результаты моделей в большем пространстве параметров и 60 попытками:

	Model	Best_Params	Best_F1	Accuracy	Precision	Recall	ROC_A
0	LGBM	{'learning_rate': 0.1, 'num_leaves': 31, 'max_...	0.654762	0.711443	0.639535	0.670732	0.7139
1	CatBoost	{'learning_rate': 0.05, 'depth': 7, 'n_estimat...	0.650000	0.721393	0.666667	0.634146	0.7401
2	RandomForest	{'n_estimators': 50, 'max_depth': None, 'min_s...	0.625767	0.696517	0.629630	0.621951	0.7302
3	HistGradientBoosting	{'learning_rate': 0.1, 'max_depth': 3, 'l2_reg...	0.621118	0.696517	0.632911	0.609756	0.7209
4	GradientBoosting	{'learning_rate': 0.2, 'n_estimators': 150, 'm...	0.615385	0.676617	0.597701	0.634146	0.7085
5	XGBoost	{'learning_rate': 0.1, 'max_depth': 7, 'n_esti...	0.602410	0.671642	0.595238	0.609756	0.7092

В ходе проведённого исследования были протестированы несколько бустинговых моделей на задаче бинарной классификации: превышает ли значение SI пороговое значение 8?

На первом этапе были обучены модели с параметрами "из коробки", что позволило получить базовый уровень (baseline) для последующего сравнения.

Далее был запущен процесс оптимизации гиперпараметров с помощью Optuna:



Сначала с 30 попытками в умеренном пространстве параметров, Затем с 60 попытками и расширенным search space для более глубокой настройки.



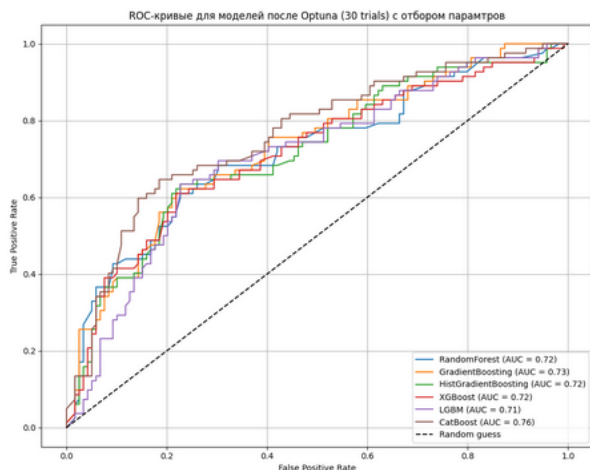
Результаты показали следующее:

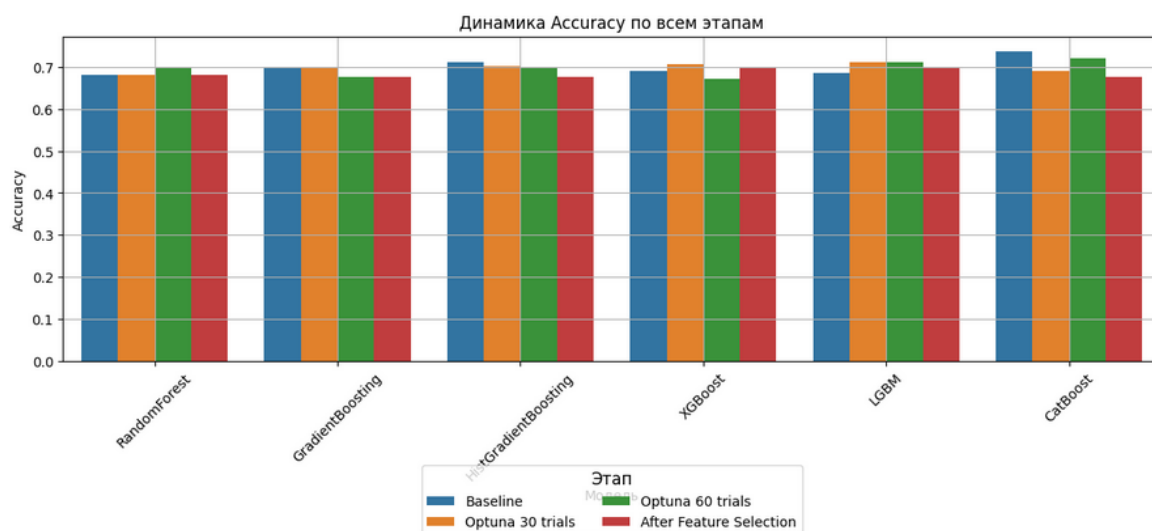
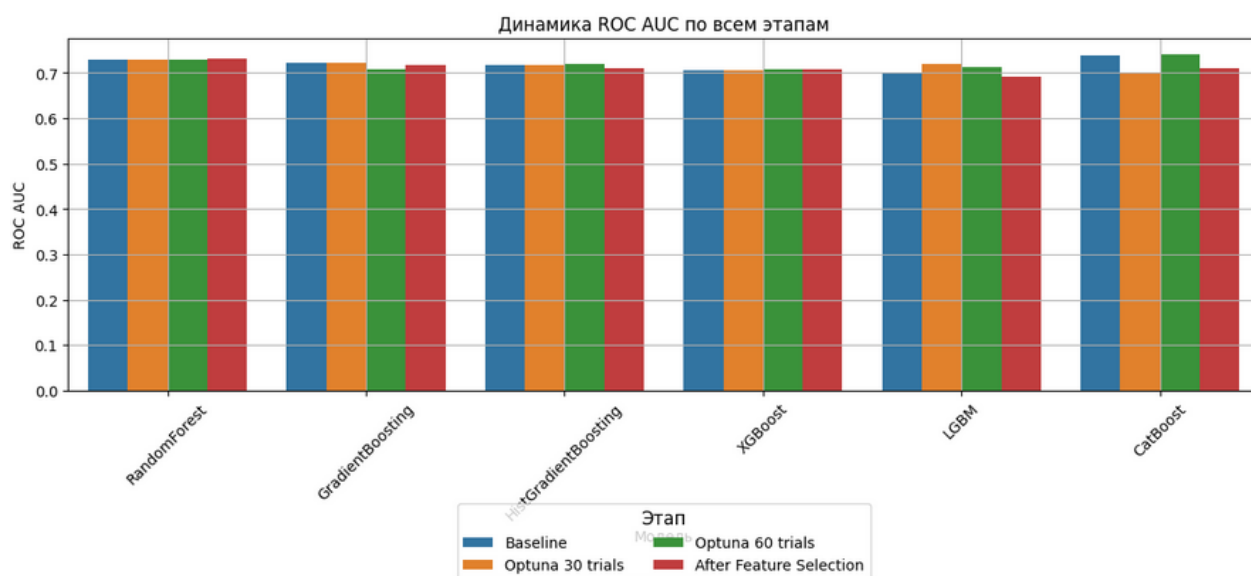
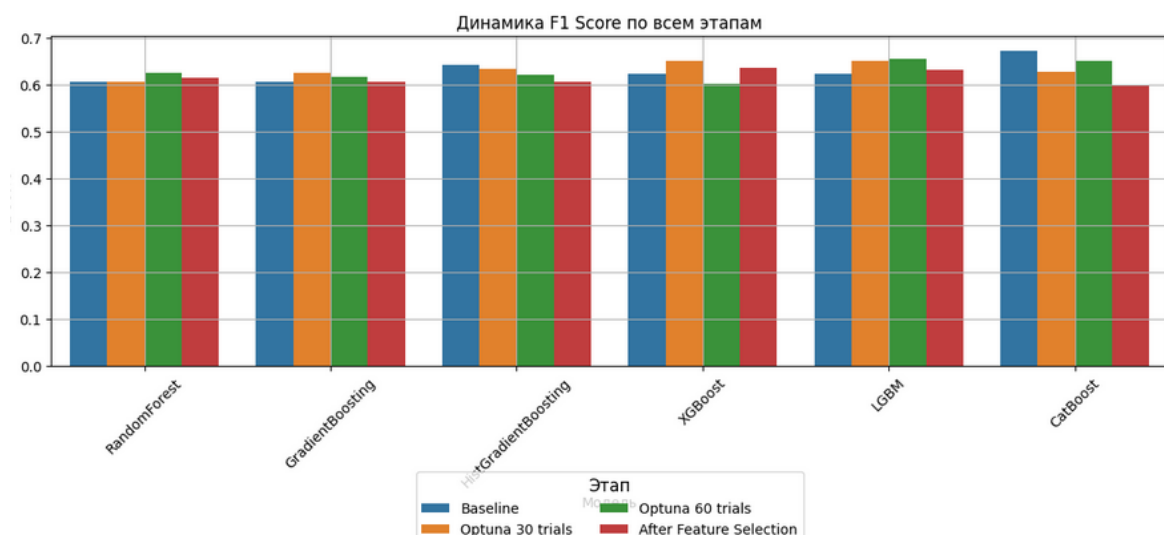
Увеличение числа trials до 60 и расширение диапазона гиперпараметров не привело к значительному улучшению метрик большинства моделей. Для некоторых моделей наблюдалось ухудшение F1 Score, например, у XGBoost и Gradient Boosting. Модели CatBoost и LightGBM остались стабильными и показали лучшие значения ROC AUC и F1 Score. RandomForest продемонстрировал умеренный прогресс, но всё ещё проигрывает бустинговым методам.

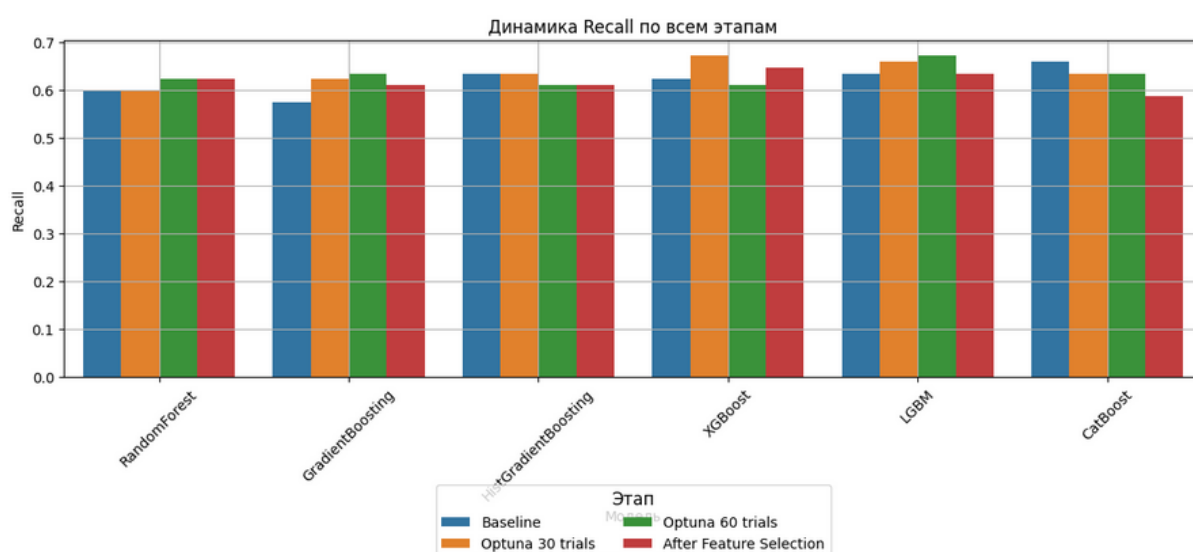
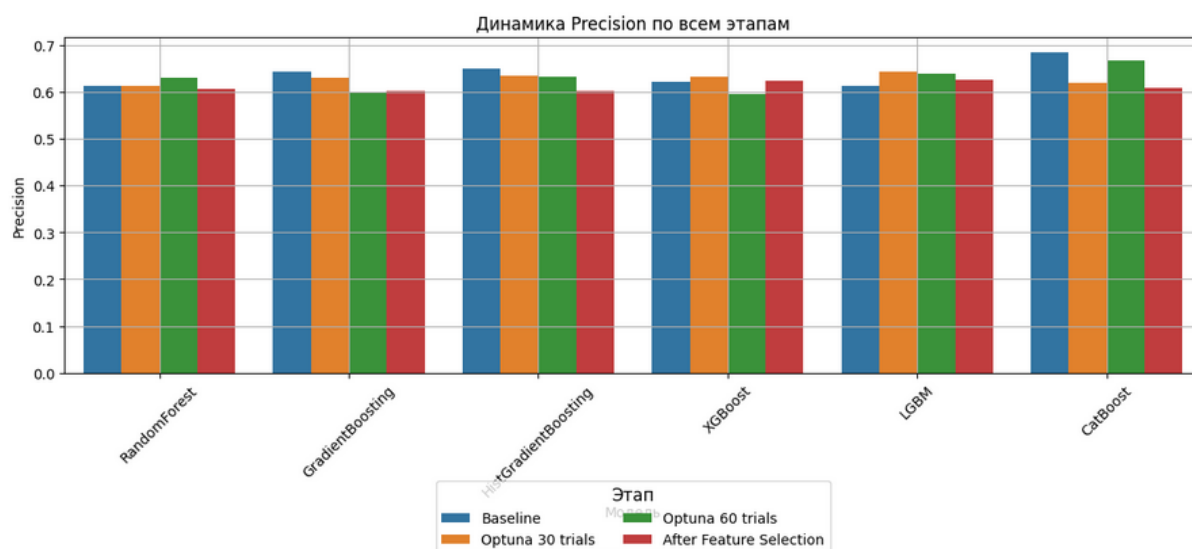
Таким образом, можно сделать следующие заключения:

Увеличение количества попыток и расширение гиперпараметрического пространства не гарантирует улучшения обобщающей способности модели, особенно если модель изначально хорошо настроена по умолчанию. CatBoost и LightGBM оказались наиболее устойчивыми и перспективными для дальнейшей работы. XGBoost и Gradient Boosting показали снижение качества на фоне усложнения параметрического пространства. Исходное пространство гиперпараметров (использовавшееся на первом запуске Optuna) оказалось более эффективным для большинства моделей. Качество данных и отбор признаков остаются ключевыми факторами для повышения точности и интерпретируемости модели.

Следующий шаг — анализ важности признаков и их отбор — позволит улучшить качество прогноза за счёт снижения влияния шума и увеличения обобщающей способности.







Сводная таблица по ROC AUC:

	Model	Baseline_AUC	Optuna_30_AUC	Optuna_60_AUC	With_Selection_AUC
0	RandomForest	0.7295	0.7295	0.7302	0.7322
1	GradientBoosting	0.7213	0.7233	0.7086	0.7181
2	HistGradientBoosting	0.7167	0.7177	0.7210	0.7098
3	XGBoost	0.7062	0.7060	0.7092	0.7084
4	LGBM	0.7021	0.7188	0.7139	0.6920
5	CatBoost	0.7388	0.6995	0.7402	0.7101

Сводная таблица по F1 Score:

	Model	Baseline_F1	Optuna_30_F1	Optuna_60_F1	With_Selection_F1
	RandomForest	0.6049	0.6049	0.6258	0.6145
	GradientBoosting	0.6065	0.6258	0.6154	0.6061
	HistGradientBoosting	0.6420	0.6341	0.6211	0.6061
	XGBoost	0.6220	0.6509	0.6024	0.6347
	LGBM	0.6228	0.6506	0.6548	0.6303
	CatBoost	0.6708	0.6265	0.6500	0.5963

Повторный запуск моделей “из коробки” подтвердил лучшими остались метрики полученные для catboost при базовом запуске. Отбор признаков и увеличение пространства параметров не привели к улучшению метрик.

Baseline метрики с параметрами:

	Model	F1 Score	ROC AUC
0	CatBoost	0.670807	0.738830
1	Hist Gradient Boosting	0.641975	0.716694
2	LightGBM	0.622754	0.702142
3	XGBoost	0.621951	0.706241
4	Gradient Boosting	0.606452	0.721306
5	Random Forest	0.604938	0.729453

Лучшая модель: CatBoost  
– F1 Score: 0.6708  
– ROC AUC: 0.7388

⚙ Параметры лучшей модели (CatBoost):  
verbose: 0  
random\_state: 42

## Глава 9

### Заключение

В рамках данной курсовой работы были решены следующие задачи:

- Проведён полный цикл EDA и предобработки данных.
- Построены и протестированы регрессионные и классификационные модели.
- Выполнено сравнение различных подходов к машинному обучению.
- Выбраны и обоснованы лучшие модели по метрикам качества.

### Основные выводы:

Сравнительная таблица моделей

МОДЕЛЬ	ЗАДАЧА	R <sup>2</sup> / F1	ROC AUC	КОММЕНТАРИЙ
XGBoost	IC <sub>50</sub>	0.34	-	Лучшая регрессия
CatBoost	CC <sub>50</sub>	0.51	-	Наивысшая точность
HistGradientBoosting	SI > медианы	0.68	0.76	Сбалансированная модель
AutoKeras	SI > медианы	0.66	0.769	Перспективно для развития

- **CatBoost** и **XGBoost** показали высокую эффективность в задачах регрессии и классификации.

- **HistGradientBoosting** демонстрирует лучший баланс между точностью и полнотой в задачах классификации.
- **AutoKeras** продемонстрировал конкурентоспособные результаты, что открывает перспективы для использования нейросетей в дальнейших работах.
- Все модели имеют ограничения, связанные с шумом и сложностью целевых переменных. Однако они позволяют эффективно ранжировать вещества по активности и селективности.