Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МИФИ»»

Институт интеллектуальных кибернетических систем

Кафедра «Криптология и Кибербезопасность»

КУРСОВАЯ РАБОТА
по теме:

ИССЛЕДОВАНИЕ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗА КЛЮЧЕВЫХ ПАРАМЕТРОВ ЛЕКАРСТВЕННЫХ СОЕДИНЕНИЙ

Автор работы Акимова Дарья Андреевна

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Разработка нового лекарственного препарата — это сложный, трудоемкий и дорогостоящий процесс, включающий такие этапы, как определение химической структуры, синтез соединения, проведение доклинических и клинических испытаний. Одним из ключевых этапов на ранних стадиях разработки является оценка биологической активности и цитотоксичности соединений, что позволяет выделить наиболее перспективных кандидатов для дальнейшего изучения.

Современные методы машинного обучения открывают новые возможности для ускорения и повышения точности данного процесса. Применение алгоритмов машинного обучения позволяет предсказывать параметры активности и безопасности химических соединений без необходимости их физического синтеза и тестирования, что существенно снижает временные и финансовые затраты.

Целью данного исследования является построение и анализ моделей машинного обучения для прогнозирования ключевых показателей эффективности химических соединений:

· IC_{50} — концентрация вещества, подавляющая активность вируса на 50%;

· CC_{50} — концентрация вещества, вызывающая гибель 50% клеток (цитотоксичность);

· SI (Selectivity Index) — индекс селективности, рассчитываемый как отношение CC_{50} к IC_{50} и характеризующий соотношение эффективности и токсичности препарата.

На основе предоставленных данных о 1000 химических соединениях, включая числовые характеристики молекул и значения целевых показателей, были построены модели регрессии и классификации, позволяющие автоматически оценивать потенциал новых соединений для использования в качестве лекарственных средств.

Для достижения поставленной цели были решены следующие задачи:

1. Провести предварительный анализ данных: изучить распределение признаков, проверить данные на наличие аномалий, пропусков и корреляций.

2. Подготовить данные к обучению моделей: выполнить нормализацию, разделить выборку на обучающую и тестовую.

3. Обучить и протестировать модели машинного обучения для решения следующих задач:

3.1 Регрессия: прогноз значений IC_{50}, CC_{50}, SI;

3.2 Классификация: определение превышает ли значение IC_{50}, CC_{50} или SI медианное значение выборки, а также превышает ли SI значение 8.

4. Сравнить качество полученных моделей на основе метрик: *MSE*, *MAE*, R^2 для регрессии; *Accuracy*, *Precision*, *Recall*, *F1-score*, *ROC-AUC* для классификации.

5. Выполнить анализ важности признаков и интерпретацию моделей.

6. Сделать выводы о применимости различных подходов и предложить рекомендации по дальнейшему использованию и улучшению моделей.

1 Анализ данных и их предобработка

1.1 Описание данных.

Для выполнения задач исследования был предоставлен датасет, содержащий информацию о 1001 химическом соединении, каждое из которых характеризуется числовыми, молекулярными дескрипторами и биологическими параметрами активности. Данные предназначены для прогнозирования эффективности и безопасности потенциальных лекарственных препаратов против вируса гриппа.

1) Целевые переменные

· IC_{50} *(Inhibitory Concentration 50%)* — концентрация вещества, подавляющая активность вируса на 50%. Чем меньше значение, тем эффективнее вещество. Более формальное определение, взятое из открытых источников: показатель эффективности лиганда при ингибирующем биохимическом или биологическом взаимодействии. IC_{50} является количественным индикатором, который показывает, сколько нужно лиганда—ингибитора для ингибирования биологического процесса на 50 %. Этот показатель обычно используется в качестве индикатора активности вещества-антагониста в фармакологических исследованиях. Формула, в которой участвует этот показатель:

$$K_i = \frac{IC_{50}}{1 + \frac{[S]}{K_m}}$$

где K_i — активность связывания ингибитора с реакционным субстратом, IC_{50} — функциональные активность ингибитора, $[S]$ — концентрация реакционного субстрата, K_m — константа Михаэлиса.

· CC_{50} *(Cytotoxic Concentration 50%)* — концентрация вещества, вызывающая гибель 50% клеток. Характеризует безопасность препарата. Более формальное определение: 50% цитотоксическая концентрация, при которой разрушается 50% клеток неинфицированного монослоя.

· *SI (Selectivity Index)* — индекс селективности, рассчитывается как отношение CC_{50} к IC_{50}. Показывает соотношение между эффективностью и токсичностью. Это безразмерная величина, которая также не может иметь отрицательных значений (поскольку считается как отношение двух неотрицательных величин).

2) Признаки (features)

Всего в датасете содержится более 200 числовых признаков, представляющих собой молекулярные дескрипторы и структурные характеристики соединений. Все признаки являются количественными и могут быть условно разделены на следующие группы:

· ***Базовые определения EState***

E-State — это численные значения, характеризующие атомы в молекуле с точки зрения их электронного состояния с учётом топологического окружения.

o *MaxAbsEStateIndex* — максимальный абсолютный E-state индекс атома — показывает структурную электроотрицательность.

o *MaxEStateIndex* — максимальный E-state индекс атома.

o *MinAbsEStateIndex* — минимальный абсолютный E-state индекс атома.

o *MinEStateIndex* — минимальный E-state индекс атома.

· ***Лекарственные характеристики***

o *qed (Quantitative Estimate of Druglikeness)* означает количественную оценку лекарственности (drug-likeness), и эту концепцию впервые представили Ричард Бикертон и его коллеги. Эмпирическая основа меры QED отражает распределение ключевых молекулярных свойств, включая молекулярную массу, логарифм коэффициента распределения вещества между двумя фазами, топологическую полярную поверхность, число доноров и акцепторов водородных связей, количество ароматических колец и вращающихся связей, а также присутствие нежелательных химических функциональных групп.

· ***Характеристики связи, электронные, зарядные***

o *SPS* — среднее количество связей на тяжелый атом. Spacial Score — это численный показатель, отражающий пространственную сложность молекулы, который используется для сравнения соединений и их ранжирования по уровню структурной "сложности". Основные параметры, которые учитываются: гибридизация атомов (sp^3-углероды), наличие стереоцентров, присутствие неароматических колец, число соседних атомов.

o *NumValenceElectrons* — общее число валентных электронов.

o *NumRadicalElectrons* — количество неспаренных электронов.

o *MaxPartialCharge* — максимальный частичный заряд атома.

o *MinPartialCharge* — минимальный частичный заряд атома.

o *MaxAbsPartialCharge* — максимальный по модулю частичный заряд атома.

o *MinAbsPartialCharge* — минимальный по модулю частичный заряд атома.

· ***Молекулярное дескрипторы***

o *MolWt* — средняя молекулярная масса соединения.

o *HeavyAtomMolWt* — средняя молекулярная масса атома, без учета водорода.

o *ExactMolWt* — точная молекулярная масса (с учётом изотопов).

· ***Плотность фингепринтов***

FpDensityMorgan1, FpDensityMorgan2, FpDensityMorgan3 — плотность моргановских фингепринтов радиусов 1, 2, 3 соответственно. Плотность равна количеству активных битов в фингерпринте, делённому на число тяжёлых атомов в молекуле. Подсчитывается общее количество уникальных атомных окружений типа Morgan (до радиуса 1, 2, 3), нормализованное по числу тяжёлых атомов в молекуле.

· ***BCUT-дескрипторы***

Различные *BCUT-дескрипторы (BCUT2D_MWHI, BCUT2D_MWLOW, BCUT2D_CHGHI, BCUT2D_CHGLO, BCUT2D_LOGPHI,*

BCUT2D_LOGPLOW, BCUT2D_MRHI, BCUT2D_MRLOW), характеризующие молекулу по массе, заряду, logP и MR.

- *BCUT2D_MWHI* — расчёты подобия свойства на определённом расстоянии от каждого атома.
- *BCUT2D_MWLOW* — минимальное собственное значение, взвешенное по атомным массам.
- *BCUT2D_CHGHI* — расчёты распределения зарядов для атомов.
- *BCUT2D_CHGLO* — минимальное собственное значение, взвешенное по зарядам Гастейгера.
- *BCUT2D_LOGPLOW* — минимальное собственное значение, взвешенное по значению логР по Криппену.
- *BCUT2D_MRLOW* — дескриптор BCUT, соответствующий наименьшему собственному значению, взвешенному по MRR (молярному рефракционному индексу) по Криппену (Crippen MRR).

· ***Топологические / топохимические дескрипторы***

- *AvgIpc* — средняя информационная энтропия коэффициентов характеристического многочлена матрицы смежности графа молекулы, построенного без учёта атомов водорода.
- *BalabanJ (Балабанский индекс J)* — мера сложности структуры. Это количественная мера сложности и разветвлённости структуры молекулы. Значение *BalabanJ* зависит от количества связей в молекуле и от распределения расстояний между атомами. Чем более разветвлённа молекула, тем меньше значение *BalabanJ*.
- *BertzCT* — индекс сложности Бертца. Отражает степень разветвлённости и цикличности структуры молекулы, а также разнообразие типов связей и атомных окружений. Чем выше значение индекса, тем сложнее структура молекулы с точки зрения топологии.
- Chi0 — нулевой порядок молекулярного связывающего дескриптора хи (χ) .

$$\chi^0 = \sum_{i=1}^{n} \frac{1}{\sqrt{\delta_i}}$$

Где: δ_i — степень (валентность) атома i в молекулярном графе (без учёта водорода); сумма берётся по всем неуглеродным атомам или по всем атомам, в зависимости от реализации; степень атома — это количество связей, которые он имеет с другими атомами.

Остальные χ-столбцы: эти дескрипторы основаны на суммировании значений, зависящих от связности атомов в молекуле. Значения могут быть: ненормализованными (n) (просто сумма) и валентными (v) (учитывают валентные свойства атомов (например, число валентных электронов)).

Индекс может быть определён для разных «уровней» или порядков, обозначаемых числом (Chi0, Chi1, ..., Chi4), что соответствует длине пути в графе.

- *HallKierAlpha* — альфа-значение Холла–Кьера для молекулы. Параметр, используемый в расчётах топологических индексов связности, таких как Chi0, Chi1 и т.д. Он служит для коррекции валентных состояний атомов, особенно в циклических структурах, чтобы улучшить корреляцию между структурой молекулы и её физико-химическими свойствами.
- *Ipc* — информационное содержание коэффициентов характеристического многочлена матрицы смежности графа молекулы, построенного без учёта атомов водорода.
- *Kappa* — дескриптор Каппа по Холлу–Кьеру.

· ***Дескрипторы приближённой молекулярной поверхности и VSA-дескрипторы.***

- *LabuteASA* — это дескриптор, используемый в хемоинформатике для оценки доступной растворителю поверхности атомов в молекуле. *LabuteASA* даёт оценку того, насколько отдельные атомы в молекуле экспонированы во внешнюю среду, то есть доступны для взаимодействия с растворителем или другими молекулами. Он назван в честь А. П. Лабут (A.P. Labute), который разработал алгоритм его расчёта в рамках работы над приближёнными

методами вычисления площади поверхности доступной для растворителя (Solvent Accessible Surface Area, SASA).

- *VSA-дескрипторы (подгруппы PEOE, SMR, SlogP, EState, VSA_EState)* — это параметры, связанные с поверхностными свойствами молекул.
  - *PEOE_VSAx*: полярные поверхности по частичным зарядам.
  - *SMR_VSAx*: по молярной рефракции
  - *SlogP_VSAx*: по коэффициенту распределения
  - *EState_VSAx, VSA_EStatex*: по E-state индексам.
- *TPSA* — это дескриптор, который приближённо оценивает площадь поверхности молекулы, занятую полярными атомами (например, атомами кислорода и азота, участвующими в образовании водородных связей).

· ***Параметры Липинского для молекул***

- *FractionCSP3* — доля sp^3-гибридизированных углеродов.
- *HeavyAtomCount* — число тяжёлых атомов. Чем больше тяжёлых атомов, тем сложнее структура соединения.
- *NHOHCount* — число NH- или OH-групп.
- *NOCount* — число атомов азота и кислорода.
- *NumAliphaticCarbocycles* — число алифатических карбоциклов в молекуле. Карбоциклы — циклы, состоящие только из атомов углерода. Алифатические — это циклы, которые не являются ароматическими, то есть имеют хотя бы одну неароматическую связь.
- *NumAliphaticHeterocycles* — число алифатических гетероциклов в молекуле. Гетероциклы — циклы, в состав которых входят разные атомы, например: C + N, C + O, C + S и т.д.
- *NumAliphaticRings* — число алифатических циклов в молекуле.
- *NumAromaticCarbocycles* — число ароматических карбоциклов в молекуле.
- *NumAromaticHeterocycles* — число ароматических гетероциклов в молекуле.

- *NumAromaticRings* — число ароматических циклов в молекуле.
- *NumHAcceptors* — количество атомов в молекуле, которые могут принимать водородные связи.
- *NumHDonors* — число доноров водородных связей; количество атомов водорода, которые могут участвовать в образовании водородных связей.
- *NumHeteroatoms* — количество всех атомов в молекуле, кроме углерода и водорода.
- *NumRotatableBonds* — количество вращающихся (ротамерных) связей в молекуле. Обычно одинарные связи между двумя насыщенными атомами. Чем больше число вращающихся связей, тем более гибкой является молекула.
- *NumSaturatedCarbocycles* — количество циклических структур, состоящих только из углерода, в которых нет ароматических связей (то есть все связи — одинарные).
- *NumSaturatedHeterocycles* — количество гетероциклов (циклов с атомами, отличными от углерода), в которых нет ароматических связей.
- *NumSaturatedRings* — общее число всех насыщенных циклов в молекуле.
- *RingCount* — общее число циклов в молекуле.

· ***Параметры, рассчитанные по методу Криппена***

- *MolLogP* - коэффициент распределения вещества между октанолом и водой, обозначаемый как LogP. Этот метод использует атомную декомпозицию, предложенную С. Уилдманом и Г.М. Криппеном. LogP - логарифм отношения растворимости вещества в октаноле к его растворимости в воде. Отражает липофильность (жирорастворимость) молекулы.
- *MolMR* - молекулярный рефрактивный индекс (MR), атомно-основанный расчёт молярного рефракционного объёма при 20 °C. Связан с поляризуемостью молекулы и используется для оценки: липофильности, поверхностных свойств и влияния на проницаемость через мембраны.

- ***Дескрипторы фрагментов***
  - *fr_Al_COO, fr_Ar_COO, fr_COO, fr_COO2* – карбоновые кислоты.
  - *fr_Al_OH, fr_Al_OH_noTert, fr_Ar_OH* – гидроксильные группы.
  - *fr_Ar_N, fr_Nhpyrrole* – атомы азота.
  - *fr_Ar_NH, fr_NH0, fr_NH1, fr_NH2, fr_Ndealkylation1, fr_Ndealkylation2* – амины.
  - *fr_C_O, fr_C_O_noCOO* – карбонильные кислороды.
  - *fr_C_S* – тиокарбонильных групп.
  - *fr_HOCCN* – третичный алкил.
  - *fr_Imine* – имины.
  - *fr_N_O* – гидроксиламины.
  - *fr_SH* – тиоловые группы.
  - *fr_aldehyde* – альдегиды.
  - *fr_alkyl_carbamate* — число алкилкарбаматов (подверженных гидролизу).
  - *fr_alkyl_halide* — число алкилгалогенидов.
  - *fr_allylic_oxid* — число сайтов аллильного окисления (кроме стероидных диенонов).
  - *fr_amide* — число амидов.
  - *fr_amidine* — число амидиновых групп.
  - *fr_aniline* — число анилинов.
  - *fr_aryl_methyl* — число арилметильных сайтов для гидроксилирования.
  - *fr_azide* — число азида.
  - *fr_azo* — число азо-групп.
  - *fr_barbitur* — число барбитуровых групп.
  - *fr_benzene* — число бензольных колец.
  - fr_benzodiazepine — число бензодиазепинов без дополнительных конденсированных колец.
  - *fr_bicyclic* — наличие бициклической структуры.

- *fr_diazo* — число диазо-групп.
- *fr_dihydropyridine* — число дигидропиридинов.
- *fr_epoxide* — число эпоксидных колец.
- *fr_ester* — число эфиров карбоновых кислот.
- *fr_ether* — число атомов кислорода в простых эфирах (включая феноксиды).
- *fr_furan* — число фурановых колец.
- *fr_guanido* — число гуанидиновых групп.
- *fr_halogen* — число галогенов.
- *fr_hdrzine* — число гидразиновых групп.
- *fr_hdrzone* — число гидразоновых групп.
- *fr_imidazole* — число имидазольных колец.
- *fr_imide* — число имидных групп.
- *fr_isocyan* — число изоцианатов.
- *fr_isothiocyan* — число изотиоцианатов.
- *fr_ketone* — число кетонов.
- *fr_ketone_Topliss* — число кетонов, исключая диарил-, α, β-ненасыщенные диеноны и кетоны с гетероатомами у α-углерода.
- *fr_lactam* — число β-лактамов.
- *fr_lactone* — число циклических эфиров (лактонов)
- *fr_methoxy* — число метоксигрупп –OCH_3.
- *fr_morpholine* — число морфолиновых колец.
- *fr_nitrile* — число нитрилов.
- *fr_nitro* — число нитрогрупп.
- *fr_nitro_arom* — число нитрогрупп в бензольном кольце.
- *fr_nitro_arom_nonortho* — число нитрогрупп в бензольном кольце, кроме орто-положений.
- *fr_nitroso* — число нитрозогрупп (исключая NO_2).
- *fr_oxazole* — число оксазольных колец.
- *fr_oxime* — число оксимных групп.

- *fr_para_hydroxylation* — число парагидроксильных групп.
- *fr_phenol* — число фенолов.
- *fr_phenol_noOrthoHbond* — число фенольных OH, исключая заместители с внутримолекулярной водородной связью в орто-положении.
- *fr_phos_acid* — число фосфорных кислот.
- *fr_phos_ester* — число фосфорных эфиров.
- *fr_piperdine* — число пиперидиновых колец.
- *fr_piperzine* — число пиперазиновых колец.
- *fr_priamide* — число первичных амидов.
- *fr_prisulfonamd* — число первичных сульфонамидов.
- *fr_pyridine* — число пиридиновых колец.
- *fr_quatN* — число четвертичных атомов азота.
- *fr_sulfide* — число тиоэфиров.
- *fr_sulfonamd* — число сульфонамидов.
- *fr_sulfone* — число сульфоновых групп.
- *fr_term_acetylene* — число терминальных ацетиленов.
- *fr_tetrazole* — число тетразольных колец.
- *fr_thiazole* — число тиазольных колец
- *fr_thiocyan* — число тиоцианатов.
- *fr_thiophene* — число тиофеновых колец
- *fr_unbrch_alkane* — число неразветвлённых алканов минимум из 4 атомов (исключая галогенированные).
- *fr_urea* — число мочевинных групп.

1.2 Первичный анализ данных.

В рамках первичного (базового) анализа данных были проведены работы, направленные на ознакомление с датасетом и устранение его очевидных недочётов, включая:

- Наличие пропущенных значений;
- Присутствие дубликатов.

В ходе анализа были выявлены 3 строки, содержащие пропуски. Признаки, в которых встречались пропуски, следующие:

- *MaxPartialCharge*
- *MinPartialCharge*
- *MaxAbsPartialCharge*
- *MinAbsPartialCharge*
- *BCUT2D_MWHI*
- *BCUT2D_MWLOW*
- *BCUT2D_CHGHI*
- *BCUT2D_CHGLO*
- *BCUT2D_LOGPHI*
- *BCUT2D_LOGPLOW*
- *BCUT2D_MRHI*
- *BCUT2D_MRLOW*

Поскольку количество строк с пропущенными значениями составило менее 0,3 % от общего объема данных, было принято решение об их удалении. Это позволило избежать потери достоверности результатов и упростило дальнейшую работу с данными. В результате данного этапа очистки в наборе осталось 998 строк.

В ходе анализа также были выявлены 32 дубликата. Наличие повторяющихся записей может привести к следующим проблемам:

- Переобучению модели, так как она будет чаще сталкиваться с одинаковыми примерами;
- Получению завышенных или недостоверных метрик на тестовой выборке, поскольку дубликаты могут попадать как в обучающую, так и в тестовую выборки.

С учётом этого было принято решение удалить все дублирующиеся строки. После их исключения окончательный размер датасета составил 966 строк.

При оценке распределения данных каждого из признаков было выявлено, что наличествуют такие, которые не несут в себе информации в виде слабого распределения данных (преобладает одно значение, либо в столбце это значение единственно). Также с самого начала стало понятно, что колонка *Unnamed: 0* является индексом и не несет в себе информацию. Следующие колонки содержали одно значение и никакое более, судя по гистограммам, поэтому тоже не представляли для нас интереса: *NumRadicalElectrons, SMR_VSA8, SlogP_VSA9, fr_Ar_COO, 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* и другие фрагментарные дескрипторы. Пример представлен на рисунке 1.

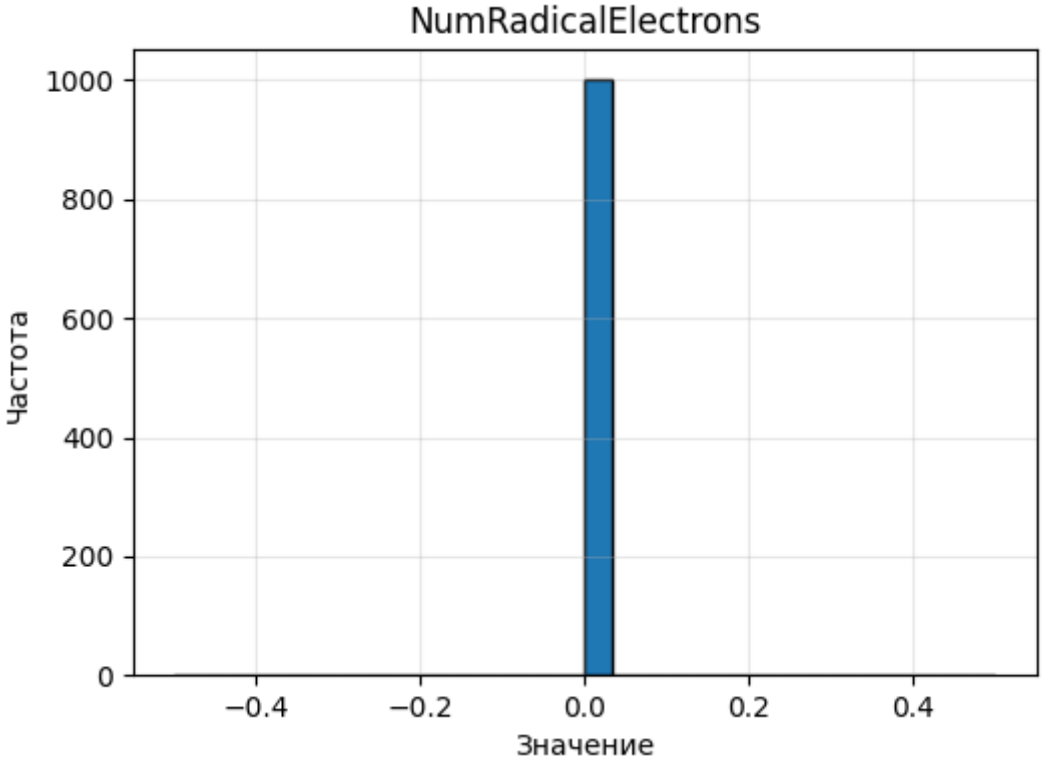

Рисунок 1 – Распределение *NumRadicalElectrons*

Очень несбаланисированное распределение данных (т.е. на гистограмме, похоже, более 95% значений - какое-то конкретное значение) наблюдалось у следующих колонок - *Ipc, SMR_VSA2, SlogP_VSA7, EState_VSA11, fr_Al_COO, fr_ArN, fr_Ar_NH, fr_COO, fr_COO2, fr_C_S, fr_HOCCN, fr_Ndealkylation1,*

fr_Ndealkylation2, fr_Nhpyrrole, fr_aldehyde, fr_alkyl_carbamate, fr_amidine, fr_azo, fr_epoxide, fr_furan, fr_guanido, fr_hdrzine, fr_hdrzone, fr_imidazole, fr_imide, fr_lactone, fr_morpholine, fr_nitrile, fr_nitro, fr_nitro_arom, fr_nitro_arom_nonortho, fr_oxazole, fr_oxime, fr_piperdine, fr_piperzine, fr_priamide, fr_pyridine, fr_quatN, fr_sulfide, fr_sulfonamd, fr_sulfone, fr_term_acetylene, fr_term_acetylene и другие фрагментарные дескрипторы. Пример представлен на рисунке 2.

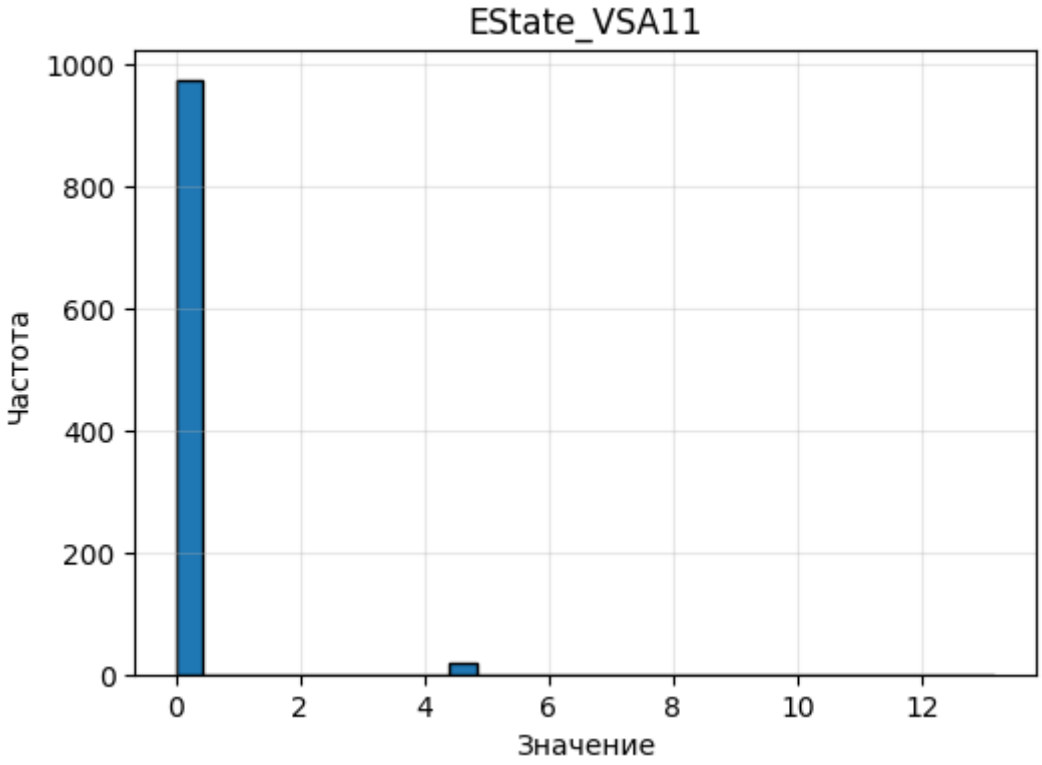

Рисунок 2 – Распределение *EState_VSA11*

Таким образом, была выполнена проверка выше перечисленных колонок (при этом проверялись абсолютно все дескрипторы фрагментов). Колонки, которые были удалены, как несущие себе единственное уникальное значение, представлены на рисунке 3.

```
['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']
```

Рисунок 3 – Колонки на удаление с единственным уникальным значением

Колонки, которые были удалены, где одно значение доминировало в большей части строк (порогом взяты были 98% строк), представлены на рисунке 4.

```
['SMR_VSA2',
 'fr_Ar_COO',
 'fr_HOCCN',
 'fr_aldehyde',
 'fr_alkyl_carbamate',
 'fr_amidine',
 'fr_azo',
 'fr_guanido',
 'fr_hdrzine',
 'fr_nitrile',
 'fr_nitro_arom',
 'fr_nitro_arom_nonortho',
 'fr_oxazole',
 'fr_sulfonamd',
 'fr_sulfone',
 'fr_term_acetylene',
 'fr_tetrazole',
 'fr_urea']
```

Рисунок 4 – Колонки на удаление, где одно уникальное значение в более, чем 98% строк

1.3 Корреляционный анализ данных.

Для более глубокого понимания структуры датасета и взаимосвязей между признаками был проведён корреляционный анализ.

1.3.1 Общий корреляционный анализ.

На первом этапе выполнен полный корреляционный анализ всех числовых столбцов датасета с использованием коэффициента корреляции Пирсона. Результаты представлены на рисунке 5.

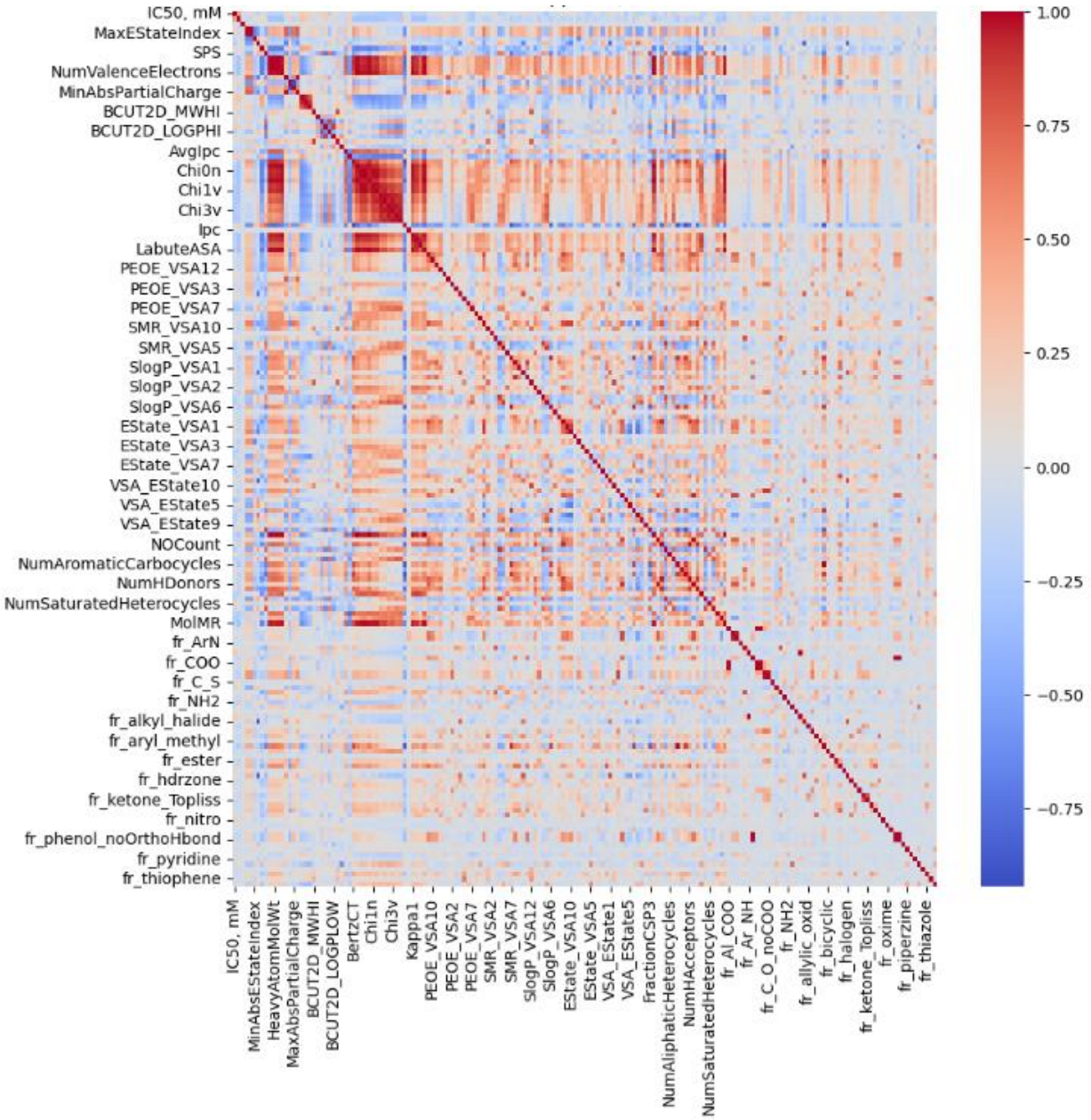

Рисунок 5 – Общая корреляционная матрица

Большая часть признаков не имеет между собой никакой корреляции, как можно видеть выше, однако наблюдаются сильно коррелирующие между собой пары (обычно линейная зависимость, как показано на рисунках 6 и 7). Такие впоследствии мы удалим, чтобы предотвратить переобучение и мультиколлинеарность. Порогом было выбрано значение 0,9, поскольку есть шанс потери данных при удалении слабее коррелирующих признаков.

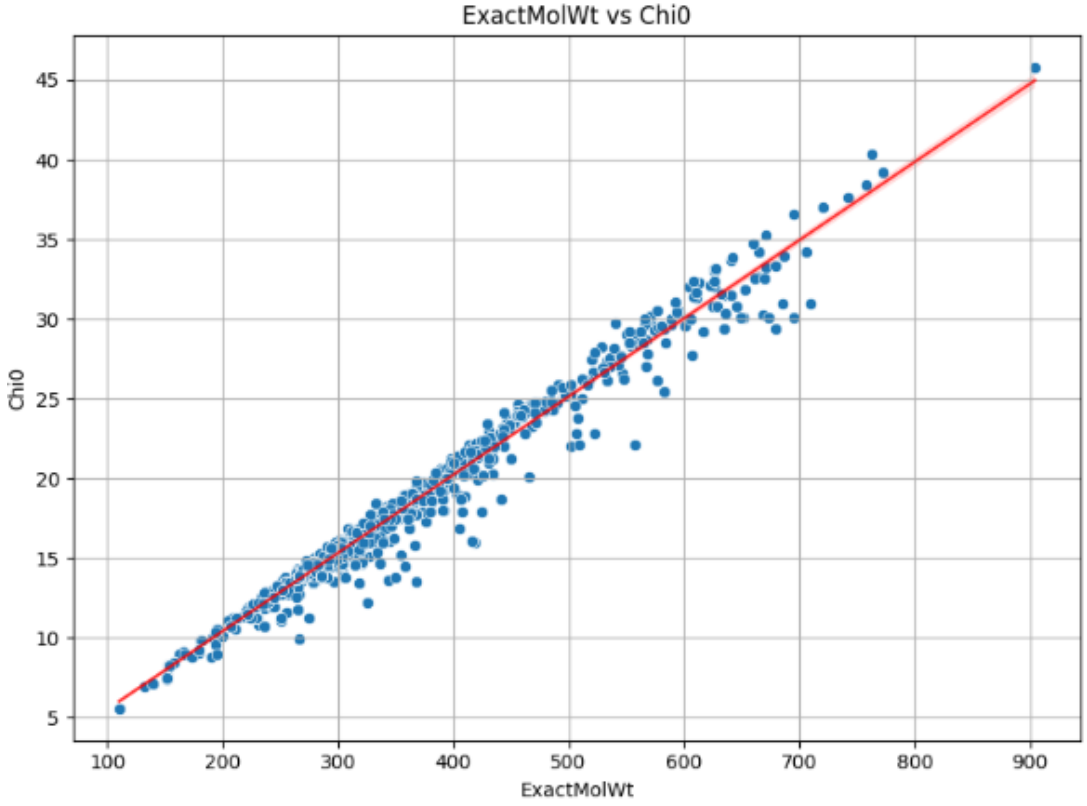

Рисунок 6 – Корреляция между *ExactMolWt* и *Chi0*

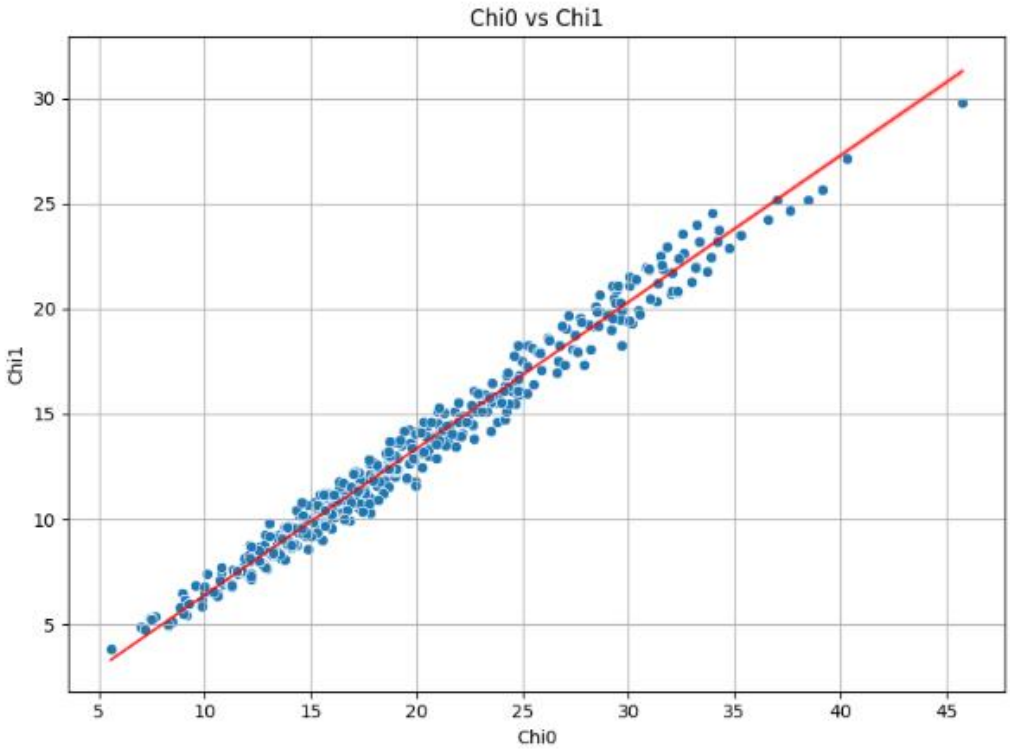

Рисунок 7 – Корреляция между *Chi0* и *Chi1*

Признаки, которые в итоге были удалены, чтобы избежать мультиколлинеарности: *MaxEStateIndex, HeavyAtomMolWt, ExactMolWt, NumValenceElectrons, MaxAbsPartialCharge, MinAbsPartialCharge, FpDensityMorgan2, FpDensityMorgan3, BCUT2D_LOGPHI, BCUT2D_LOGPLOW, BCUT2D_MRHI, AvgIpc, BertzCT, Chi0, Chi0n, Chi0v, Chi1, Chi1n, Chi1v, Chi2n, Chi2v, Chi3n, Chi3v, Chi4n, Chi4v, HallKierAlpha, Kappa1, Kappa2, Kappa3, LabuteASA, SMR_VSA1, SMR_VSA7, SMR_VSA9, SlogP_VSA10, SlogP_VSA11, SlogP_VSA12, SlogP_VSA4, SlogP_VSA5, SlogP_VSA6, TPSA, EState_VSA1, EState_VSA10, VSA_EState1, VSA_EState10, VSA_EState2, VSA_EState3, VSA_EState6, VSA_EState8, FractionCSP3, HeavyAtomCount, NHOHCount, NOCount, NumAliphaticCarbocycles, NumAliphaticRings, NumAromaticCarbocycles, NumAromaticRings, NumHAcceptors, NumHDonors, NumHeteroatoms, NumRotatableBonds, NumSaturatedCarbocycles, NumSaturatedRings, RingCount, MolMR, fr_Al_OH, fr_Al_OH_noTert, fr_Ar_N, fr_COO, fr_COO2, fr_C_O, fr_C_O_noCOO, fr_NH0,*

fr_Nhpyrrole, *fr_alkyl_halide*, *fr_benzene*, *fr_ether*, *fr_halogen*, *fr_ketone_Topliss*, *fr_phenol*, *fr_phenol_noOrthoHbond*.

1.3.2 Корреляция признаков с целевыми переменными.

Была рассмотрена корреляция признаков в отдельности с каждой из целевых переменных. На рисунках 8-10 можно видеть результаты.

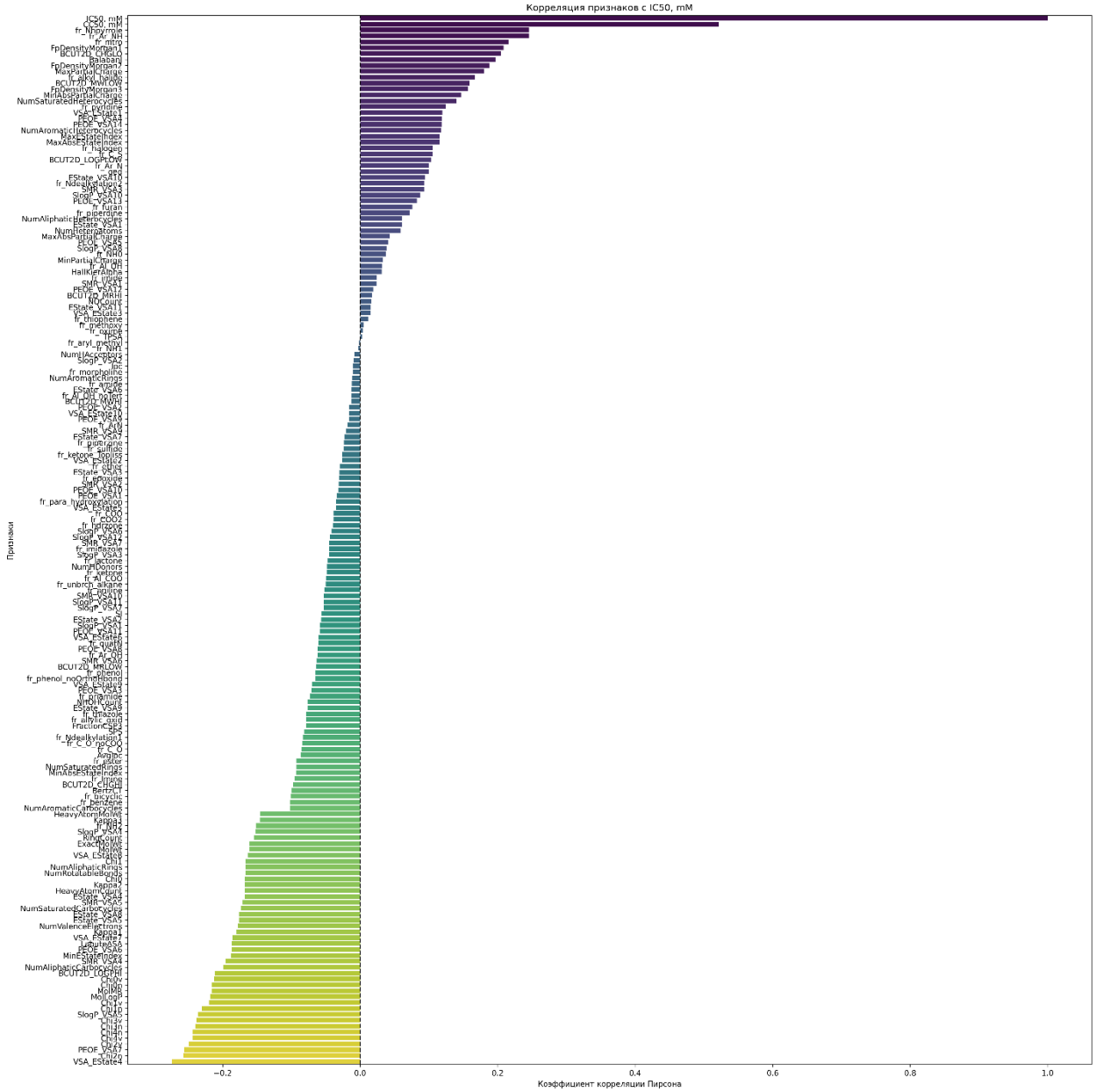

Рисунок 8 – Корреляция с целевой переменной IC_{50}

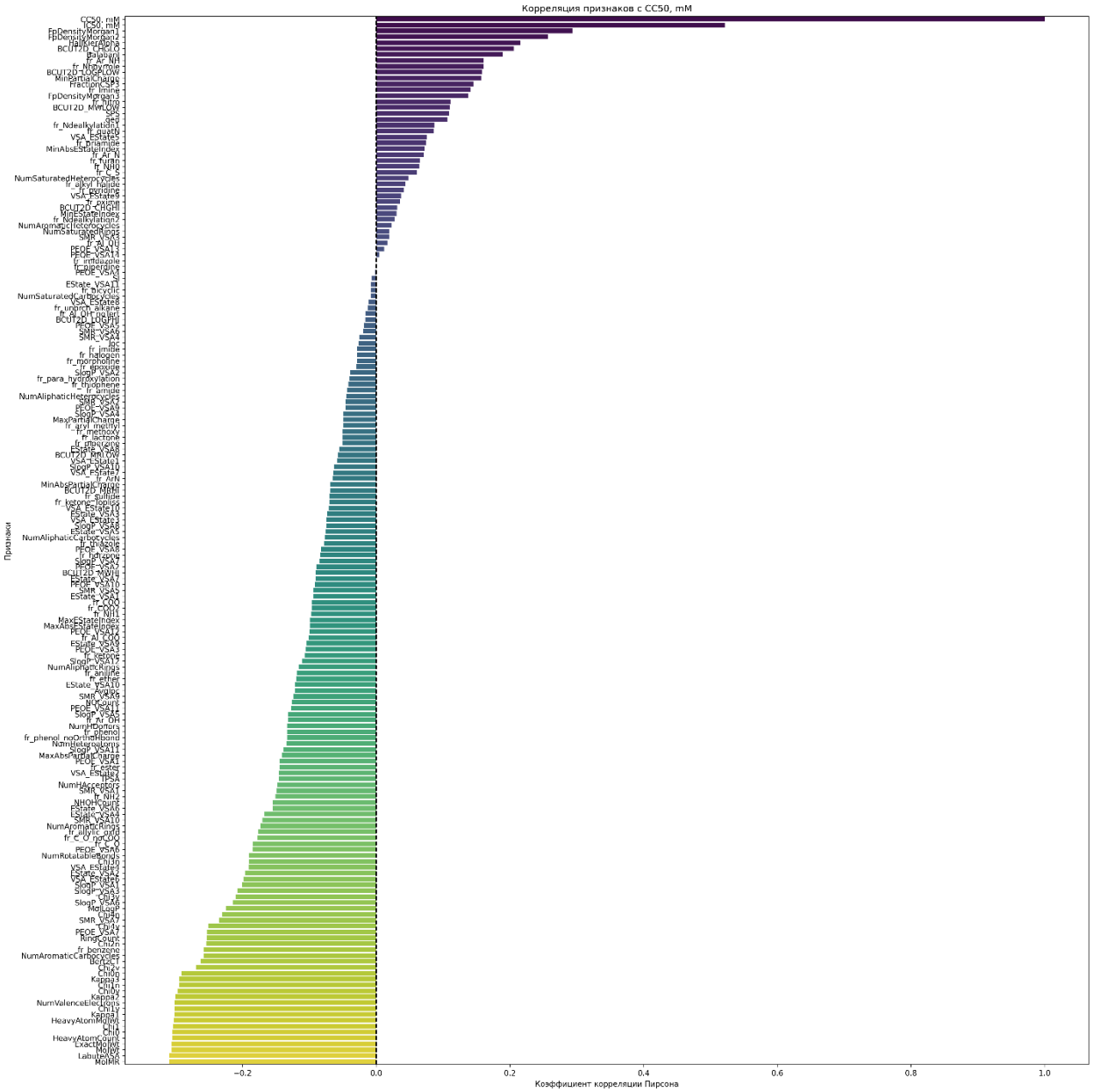

Рисунок 9 – Корреляция с целевой переменной CC_{50}

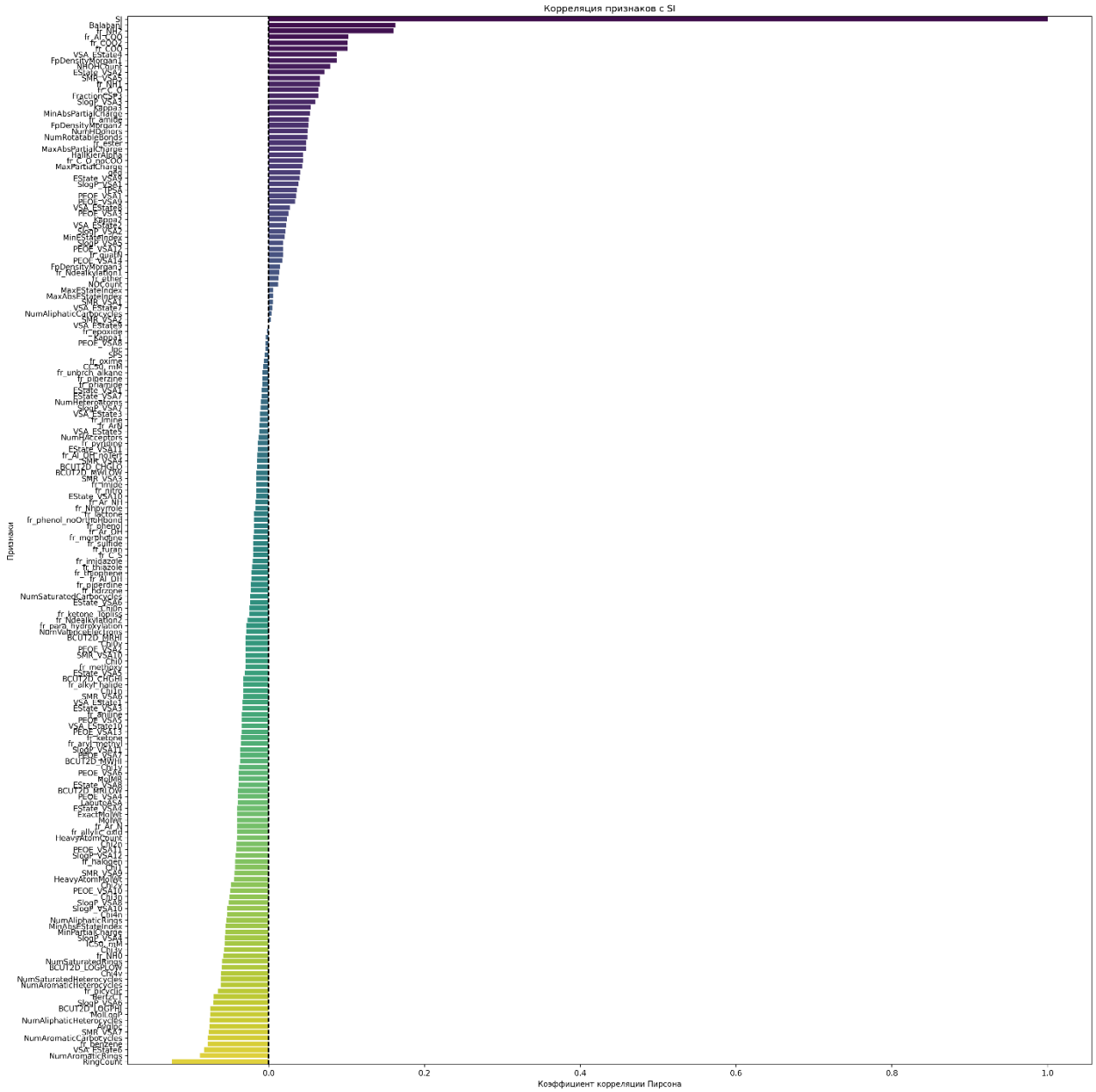

Рисунок 10 – Корреляция с целевой переменной SI

Как можно видеть на графиках, нет сильно коррелирующих признаков с целевыми переменными (в основном слабо коррелирующие, либо отсутствие корреляции). Это говорит о том, что:

· прогнозирование целевых переменных невозможно выполнить на основе одного или нескольких сильно коррелирующих признаков;

· модель машинного обучения должна учитывать комбинации признаков для выявления скрытых закономерностей.

.

1.3.3 Корреляция между целевыми переменными.

Как видно на рисунке 11, на котором отображена корреляция между целевыми переменными, между значениями IC_{50} и CC_{50} наблюдается умеренная взаимосвязь, что может свидетельствовать о связи токсичности соединений с их биологической активностью. Корреляционная зависимость между остальными парами признаков выявлена не была.

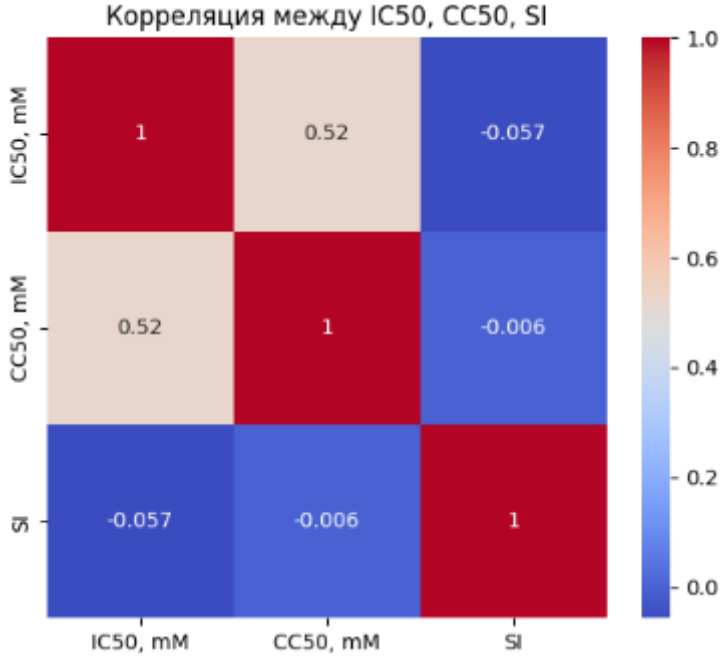

Рисунок 11 – Корреляция между целевыми переменными

1.4 Распределение целевых переменных.

Для более глубокого понимания структуры данных и подготовки к построению моделей машинного обучения был проведён анализ распределения целевых переменных: IC_{50}, CC_{50} и SI.

При изучении исходных распределений целевых показателей было установлено, что две переменные (IC_{50} и CC_{50}) имеют выраженную левостороннюю асимметрию (смещение влево). Это означает, что большинство значений сконцентрировано в области малых величин, при этом наблюдаются редкие выбросы с большими значениями.

Их распределения в исходном состоянии показаны на рисунках 12-14.

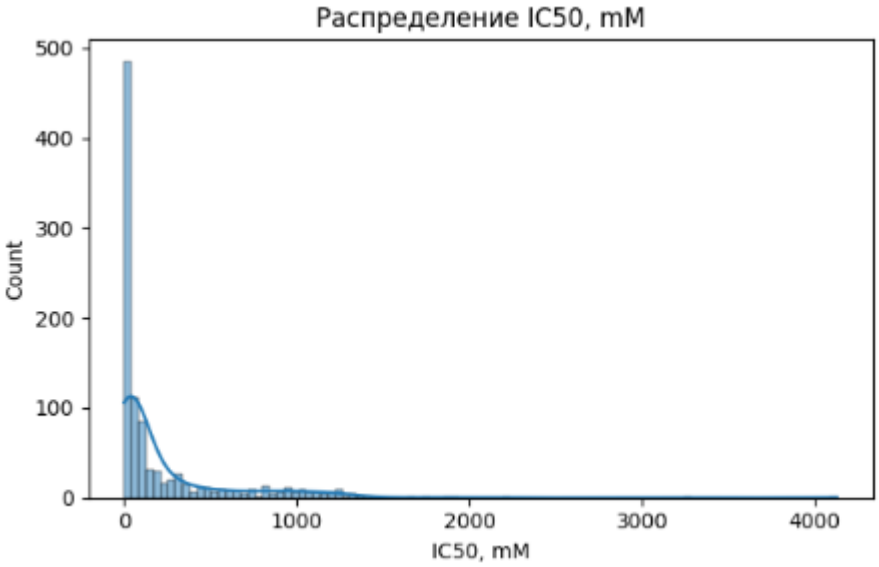

Рисунок 12 – Распределение IC_{50} в исходном состоянии

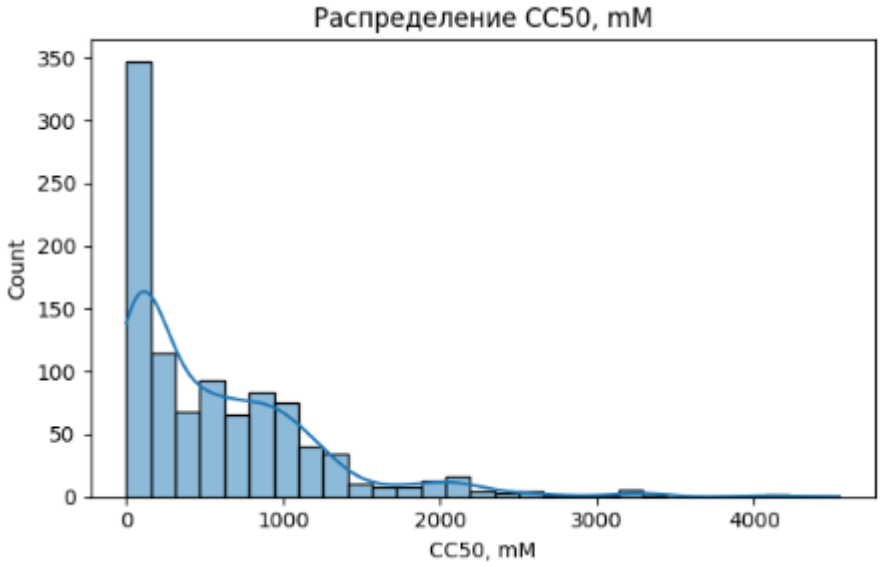

Рисунок 13 – Распределение CC_{50} в исходном состоянии

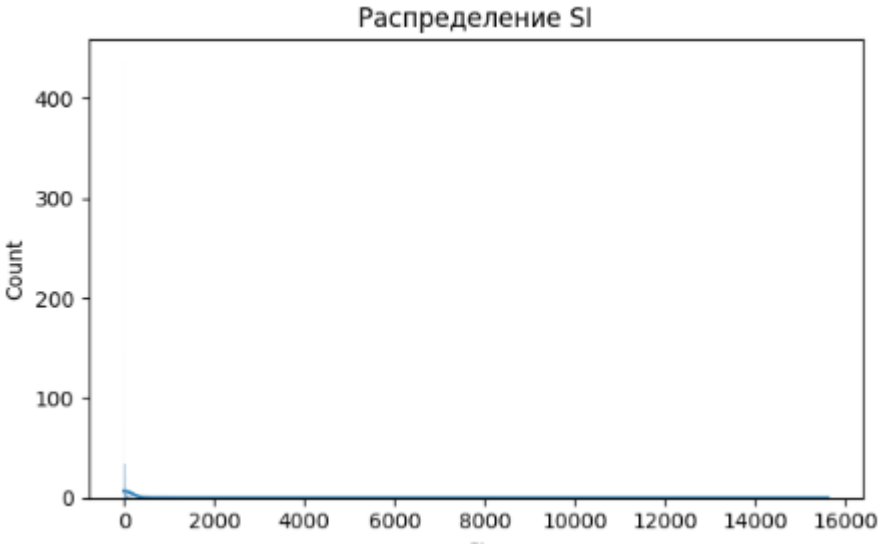

Рисунок 14 – Распределение SI в исходном состоянии

Было принято решение прологарифмировать данные, чтобы убрать влияние выбросов (или аномально больших значений), поскольку наибольшее количество данных все-таки сосредоточено в пределах границ куда меньших. Были получены следующие картины, отображенные на рисунках 15-17.

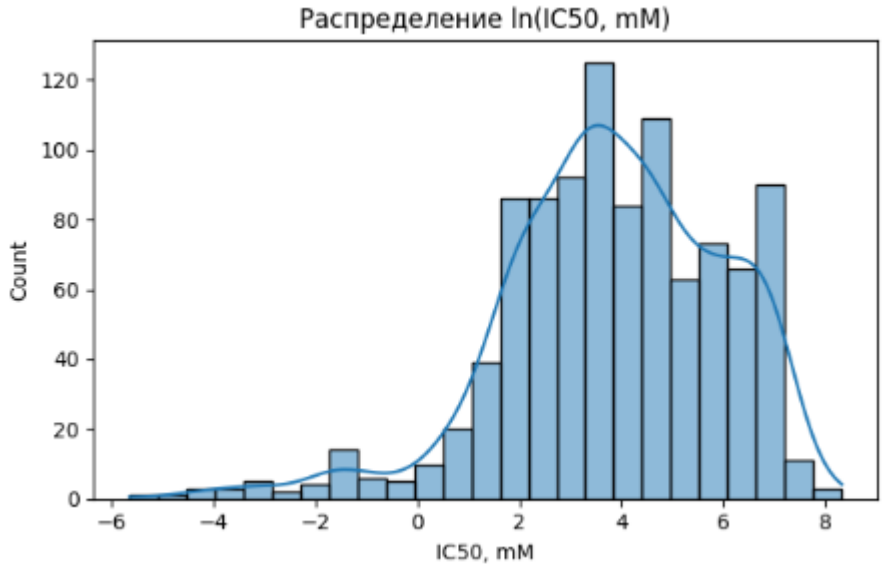

Рисунок 15 – Распределение натурального логарифма от IC_{50}

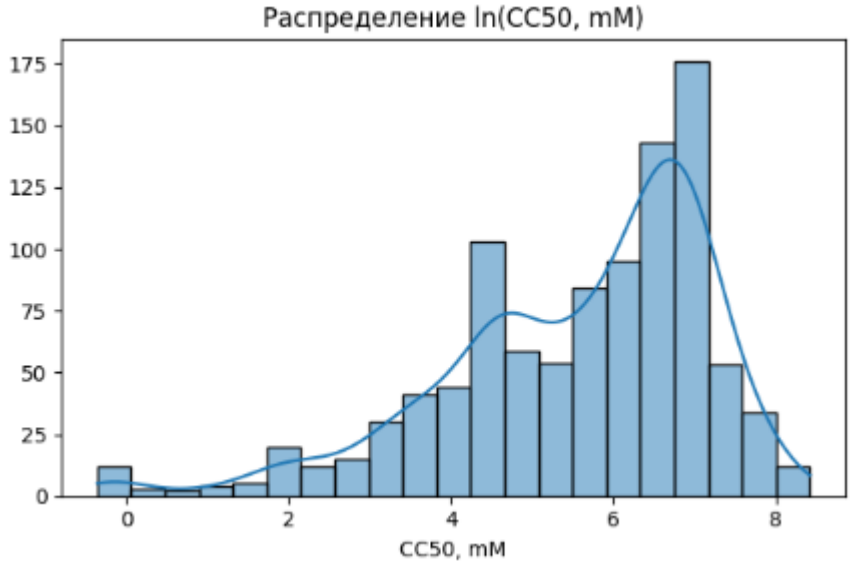

Рисунок 16 – Распределение натурального логарифма от CC_{50}

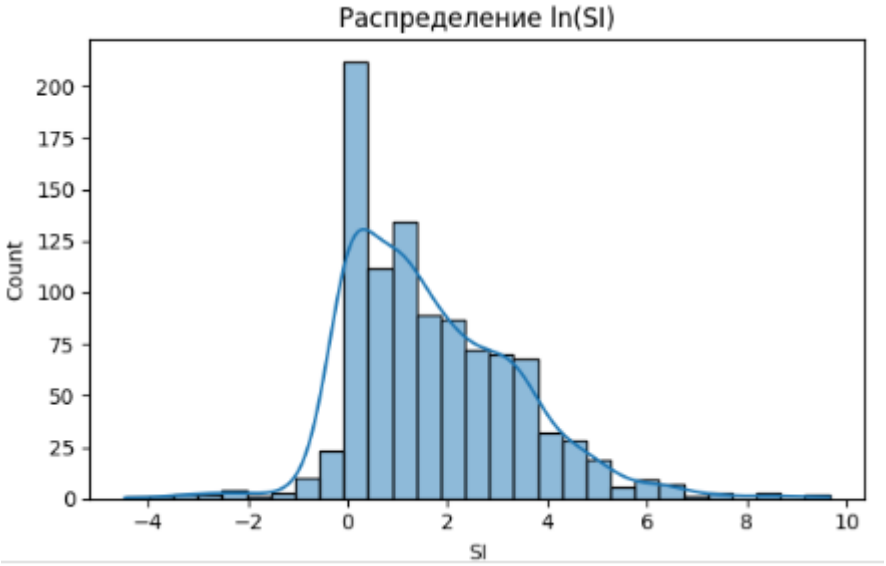

Рисунок 17 – Распределение натурального логарифма от SI

Очевидно, что логарифмирование улучшает распределение данных для целевых переменных IC_{50} и CC_{50}, что может поспособствовать нам в задачах регрессии и классификации.

1.5 Проверка на наличие выбросов.

Поскольку выбросы сильно влияют не только на общую картину, но и на качество работы моделей, было решено провести проверку на наличие

выбросов. На рисунках 18-20 показаны боксплоты для каждой из целевых переменных.

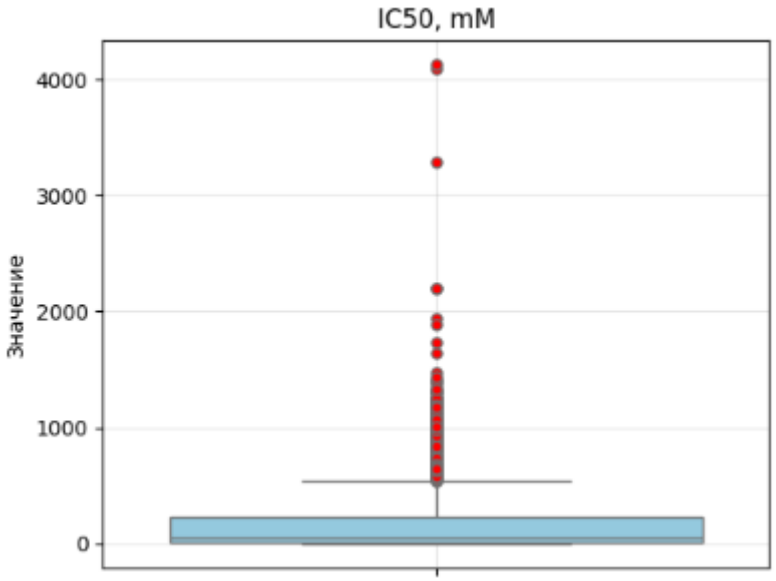

Рисунок 18 – Выбросы для IC_{50}

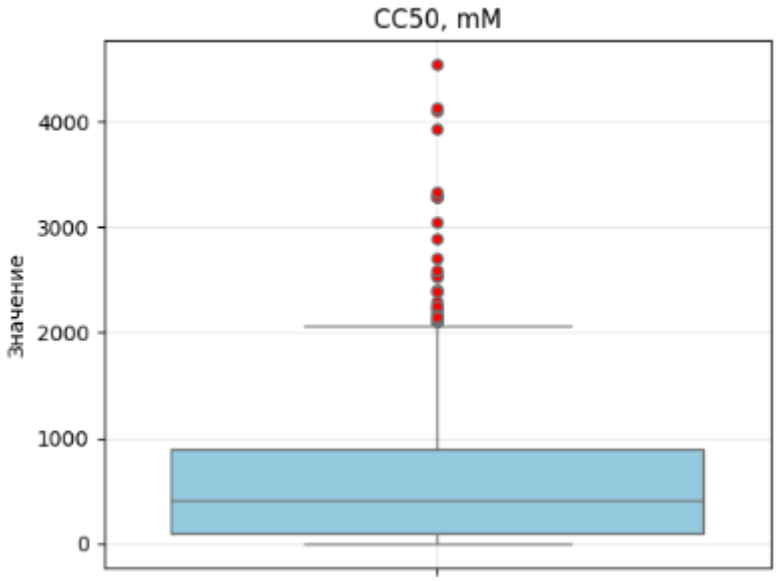

Рисунок 19 – Выбросы для CC_{50}

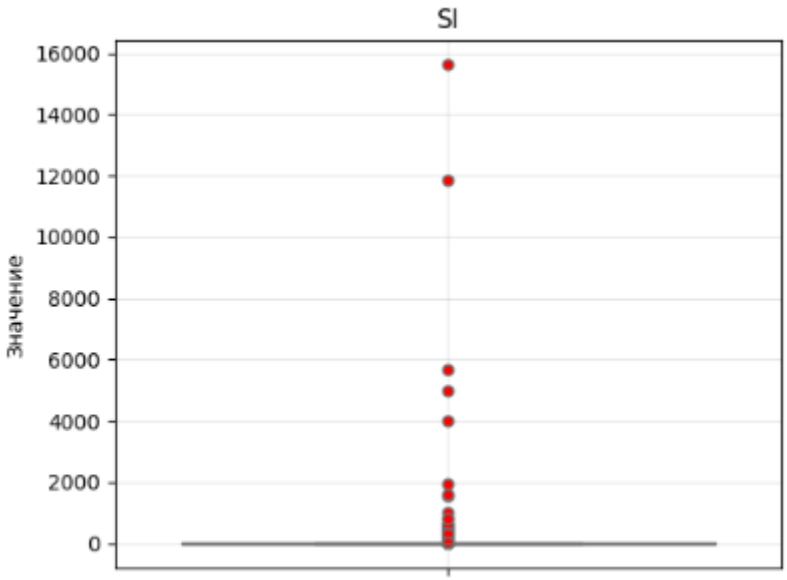

Рисунок 20 – Выбросы для SI

У первых двух целевых переменных не наблюдается большое количество выбросов. Отсеивание по 3 сигмам возможно будет достаточно для модели, тогда как у третьей это может вызвать проблемы в связи с тем, что не все красные точки могут оказаться выбросами, ибо создание малоэффективного, но очень токсичного лекарства – ситуация вполне реальная.

2 Решение задачи регрессии

Задача регрессии, как и задача классификации, решается одними методами для всех целевых переменных. Ее цель состоит в построении качественной регрессионной модели, которая могла бы с достаточной точностью предсказывать значения целевых.

Для того, чтобы решить задачу регрессии, мы произведем несколько этапов:

1. Предобработка данных (удаление выбросов).
2. Отбор информативных признаков.
3. Логарифмирование (если необходимо) целевой переменной.
4. Разделение данных на обучающую и тестовую выборки.
5. Подбор модели и гиперпараметров.
6. Сравнение моделей и выбор лучшей.

Отбор информативных признаков велся с помощью корреляции с целевой переменной, отбора *Случайным Лесом* и *Лассо*.

Для решения задач регрессии были протестированы различные модели машинного обучения, отличающиеся по сложности, скорости обучения и способности к обобщению. Для каждой модели был выполнен подбор гиперпараметров с использованием *GridSearchCV*.

Были протестированы следующие модели:

· *Линейная регрессия*. Выбрана как простая базовая модель для сравнения. Используется для оценки линейной зависимости между признаками и целевой переменной.

· *Случайный Лес* выбран благодаря своей способности эффективно обрабатывать нелинейные зависимости, устойчивости к переобучению и хорошей интерпретируемости важных признаков.

· *Градиентный бустинг* был включен в исследование за счёт её высокой точности и способности находить сложные закономерности в данных.

· *XGBoost* использовался из-за его известной эффективности и устойчивости к переобучению. Модель хорошо зарекомендовала себя в задачах с числовыми признаками и средним размером выборки.

· *CatBoost* был добавлен как современная реализация градиентного бустинга, особенно эффективная на числовых данных и с автоматической обработкой категориальных признаков (хотя в нашем случае они отсутствуют).

Оценка моделей при этом производилась с помощью следующих метрик:

· *MAE (Mean Absolute Error)* — средняя абсолютная ошибка. Эта метрика показывает среднее отклонение предсказаний от истинных значений в тех же единицах измерения, что и целевая переменная.

· *RMSE (Root Mean Squared Error)* — корень из среднеквадратичной ошибки. В отличие от MAE, RMSE более чувствительна к большим ошибкам, так как ошибки возводятся в квадрат перед усреднением. Это позволяет выявить случаи, когда модель плохо справляется с отдельными наблюдениями.

· *R^2 (коэффициент детерминации)* демонстрирует, насколько хорошо модель объясняет дисперсию целевой переменной. Значение 1 соответствует идеальной модели, 0 — константной модели, предсказывающей среднее.

· Иногда использовалась также метрика *MAPE (Mean Absolute Percentage Error)* — средняя абсолютная процентная ошибка. MAPE выражает ошибку в процентах и удобна для сравнения моделей на разных масштабах данных.

2.1 Задача регрессии для IC_{50}

2.1.1 Предобработка данных

Как было ранее указано, распределение данных у целевой переменной IC_{50} было сильно лучше, когда ее значения были логарифмированы натуральным логарифмом. Поэтому при разделении данных на

тренировочную и тестовую выборки была попытка прологарифмировать целевую переменную, однако это давало отрицательные коэффициенты детерминации при возвращении их в нормальный вид, потому для задачи регрессии логарифмирование не использовалось. Также была проведена нормализация признаков с помощью *StandardScaler*, т.к. регрессия очень чувствительна к ненормированным данным.

Помимо этого, было проведено удаление выбросов по 3δ, как показано на рисунке 21.

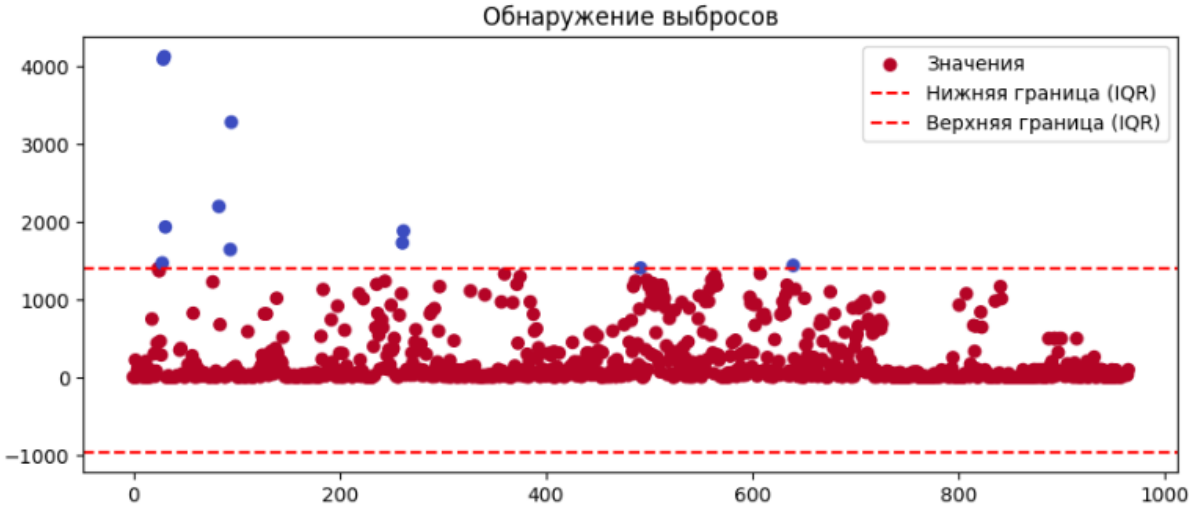

Рисунок 21 – Выбросы для IC_{50}

2.1.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,1), с помощью *Случайного Леса* (см. рисунок 22) и с помощью *Лассо* (см. рисунок 23).

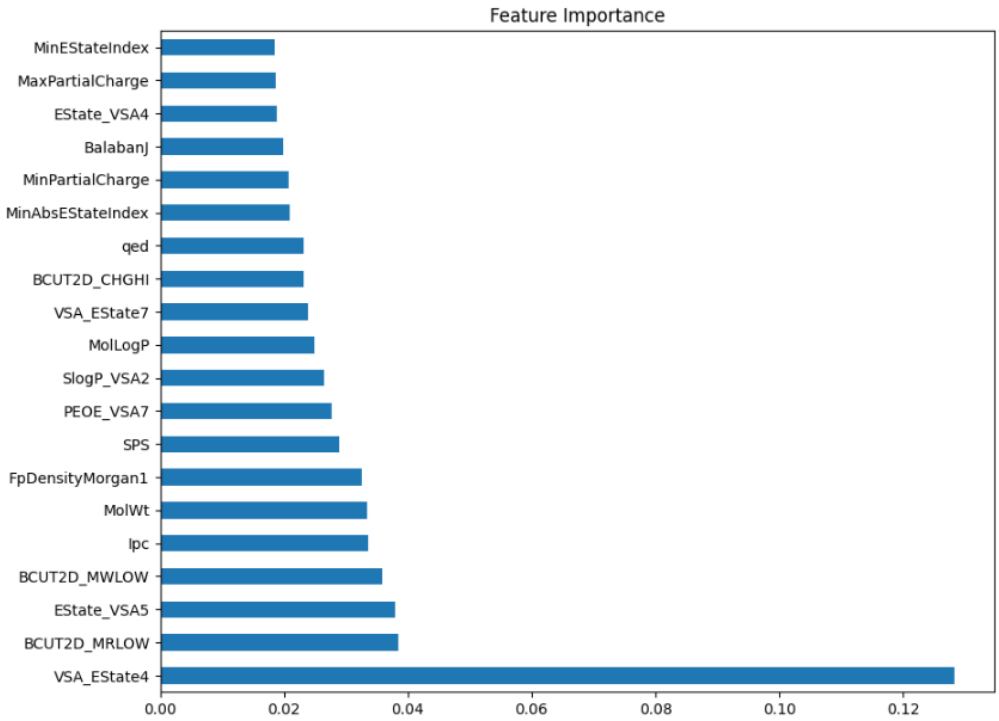

Рисунок 22 – Признаки, отобранные с помощью *Случайного Леса* для предсказания IC_{50}

```
Коэффициенты:
NumSaturatedHeterocycles    53.404739
fr_C_S                      27.130528
BCUT2D_CHGLO                26.721588
fr_pyridine                 14.144515
VSA_EState5                 11.252543
                               ...
fr_ketone                  -20.926533
fr_priamide                -21.781413
SMR_VSA10                  -23.936193
fr_NH2                     -26.211323
MinEStateIndex             -39.871689
Length: 94, dtype: float64
Оставлено признаков: 32
```

Рисунок 23 – Признаки, отобранные с помощью *Лассо* для предсказания IC_{50}

2.1.3 Подбор модели и гиперпараметров

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 24. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 212.3652, RMSE: 306.3251, R²: 0.1697, MAPE: 64.76

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 5, 'min_samples_split': 10, 'n_estimators': 100}
MAE: 210.5250, RMSE: 299.2494, R²: 0.2077, MAPE: 82.38

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 211.7451, RMSE: 312.0990, R²: 0.1381, MAPE: 76.34

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 207.6385, RMSE: 309.3003, R²: 0.1535, MAPE: 77.18

Обучение модели: CatBoost
Лучшие параметры: {'depth': 4, 'iterations': 100, 'l2_leaf_reg': 7, 'learning_rate': 0.1}
MAE: 200.8034, RMSE: 293.9288, R²: 0.2356, MAPE: 70.29
```

Рисунок 24 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания IC_{50}

На данных, признаки которых были отобраны с помощью *Случайного Леса*, были получены результаты, отображенные на рисунке 25. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 226.6706, RMSE: 321.2889, R²: 0.0866, MAPE: 128.81

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 5, 'min_samples_split': 5, 'n_estimators': 100}
MAE: 219.5996, RMSE: 304.2463, R²: 0.1810, MAPE: 97.92

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 221.4564, RMSE: 312.2427, R²: 0.1374, MAPE: 84.04

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 217.3866, RMSE: 323.4699, R²: 0.0742, MAPE: 77.28

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 300, 'l2_leaf_reg': 1, 'learning_rate': 0.01}
MAE: 201.2341, RMSE: 299.8483, R²: 0.2045, MAPE: 104.03
```

Рисунок 25 - Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания IC_{50}

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 26. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 201.6430, RMSE: 294.4942, R²: 0.2326, MAPE: 52.32

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 5, 'min_samples_split': 10, 'n_estimators': 100}
MAE: 214.3026, RMSE: 298.6213, R²: 0.2110, MAPE: 84.25

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 215.7872, RMSE: 309.4109, R²: 0.1529, MAPE: 98.87

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 211.0101, RMSE: 310.4438, R²: 0.1473, MAPE: 108.98

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.05}
MAE: 188.5646, RMSE: 284.5875, R²: 0.2834, MAPE: 61.71
```

Рисунок 26 - Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания IC_{50}

2.1.4 Выводы

Лучше всего себя показала модель *CatBoost*, обученная на данных, признаки которых отбирались с помощью корреляции. Гиперпараметры этой модели следующие:

- *depth*: 8
- *iterations*: 100
- *l2_leaf_reg*: 1
- *learning_rate*: 0,05

Метрики, которые были получены при использовании этой модели:

· *MAE* = 188,5646 мМ — средняя абсолютная ошибка прогнозирования целевой переменной, что говорит о том, что в среднем модель ошибается на ~188,5646 мМ.

· *RMSE* = 284,5875 мМ — чувствительна к большим ошибкам, и её значение немного выше MAE, что указывает на наличие отдельных неточных предсказаний, но не критичных.

· R^2 = 0,2834 — модель объясняет около 28% дисперсии целевой переменной. Это значение нельзя назвать высоким, но учитывая сложность задачи и количество данных (около 1000 образцов), его можно считать удовлетворительным.

· *MAPE* = 61,71% — довольно высокий процент ошибки, особенно если важна точность предсказания для конкретных соединений. Однако в условиях слабых корреляций между признаками и целевой переменной такой результат выглядит оправданным.

Однако, следует учитывать, что:

· Для повышения качества модели рекомендуется увеличить объём данных, например, за счёт сбора дополнительных экспериментальных значений.

· Также возможно улучшение за счёт расширения набора признаков: добавления новых дескрипторов.

· Если точность прогноза критически важна, можно рассмотреть обращение к специалистам в данной области для дальнейших действий и возможных улучшений.

2.2 Задача регрессии для CC_{50}

2.2.1 Предобработка данных

Как было ранее указано, распределение данных у целевой переменной CC_{50} было сильно лучше, когда ее значения были логарифмированы натуральным логарифмом. Поэтому при разделении данных на тренировочную и тестовую выборки была попытка прологарифмировать целевую переменную, однако это давало отрицательные коэффициенты детерминации при возвращении их в нормальный вид, потому для задачи регрессии логарифмирование не использовалось. Также была проведена

нормализация признаков с помощью *StandardScaler*, т.к. регрессия очень чувствительна к ненормированным данным.

Помимо этого, было проведено удаление выбросов по 3δ снизу и по 2000 сверху, как показано на рисунке 27. Верхний предел был изменен в связи с улучшением метрик моделей. Была предпринята попытка провести верхнюю черту по значению 1000, так как обсуждалось, что значения выше этого являются выбросами, однако это ухудшило предсказание моделей.

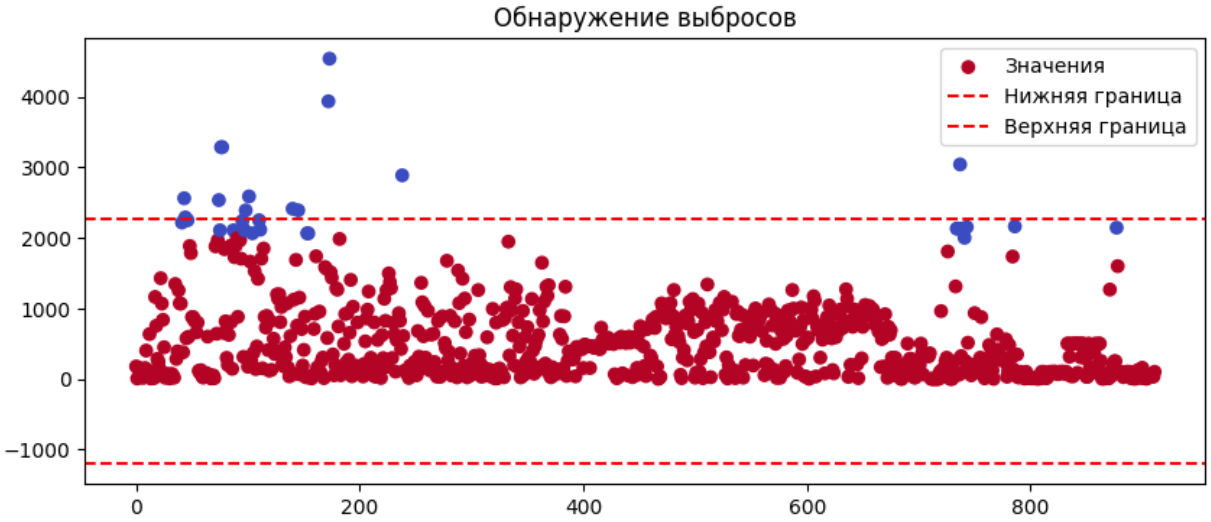

Рисунок 27 – Выбросы для CC_{50}

2.2.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,15), с помощью *Случайного Леса* (см. рисунок 28) и с помощью *Лассо* (см. рисунок 29).

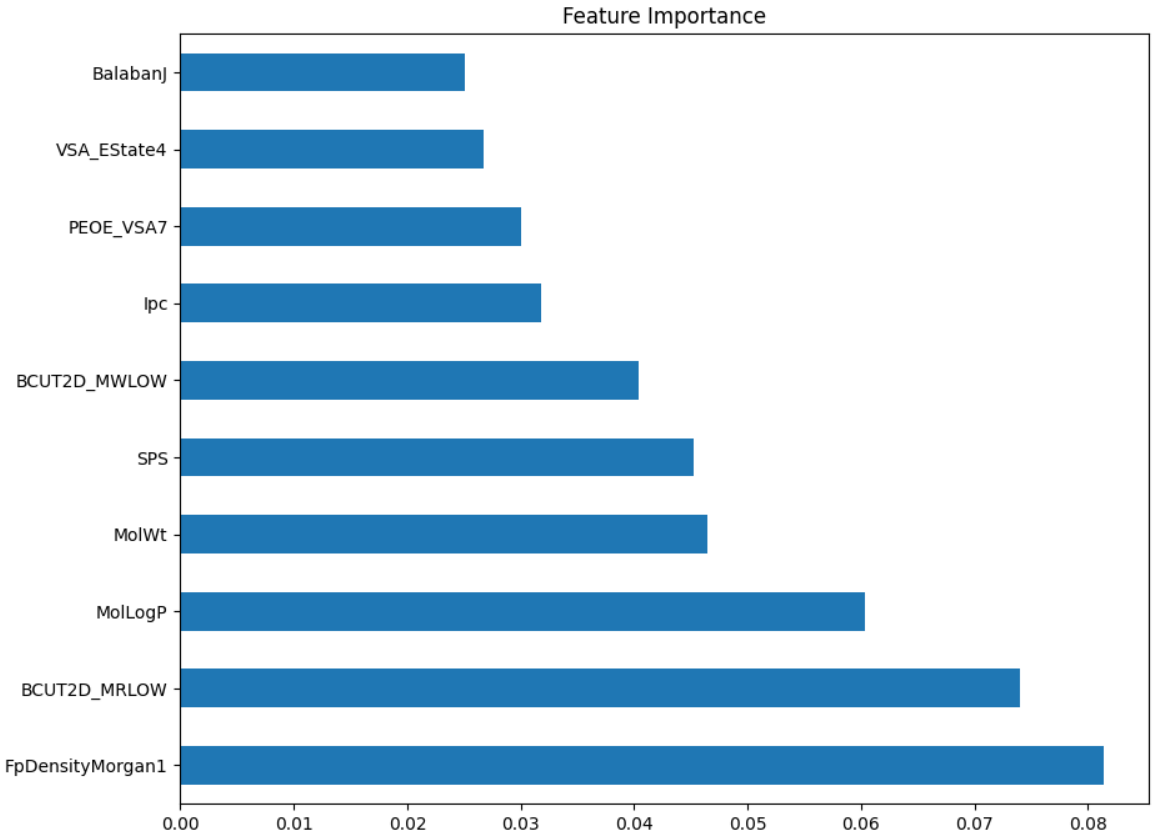

Рисунок 28 – Признаки, отобранные с помощью *Случайного Леса* для предсказания CC_{50}

```
Коэффициенты:
FpDensityMorgan1            33.594354
NumSaturatedHeterocycles    24.955382
fr_Imine                    23.906844
fr_quatN                    16.110507
fr_Ndealkylation1           14.181965
                               ...
PEOE_VSA6                  -19.625878
fr_allylic_oxid            -27.084010
PEOE_VSA7                  -31.482422
MolWt                      -51.509646
fr_NH2                     -76.992372
Length: 94, dtype: float64
Оставлено признаков: 16
```

Рисунок 29 – Признаки, отобранные с помощью *Лассо* для предсказания CC_{50}

2.2.3 Подбор модели и гиперпараметров

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 30. Как можно видеть,

лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 352.3964, RMSE: 426.4211, R²: 0.2138

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 100}
MAE: 297.6439, RMSE: 371.4864, R²: 0.4033

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 296.0416, RMSE: 376.6616, R²: 0.3866

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 331.4436, RMSE: 401.6242, R²: 0.3026

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.1}
MAE: 274.5404, RMSE: 358.6152, R²: 0.4439
```

Рисунок 30 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания CC_{50}

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 31. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 320.0071, RMSE: 389.1207, R²: 0.3453

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': None, 'min_samples_split': 5, 'n_estimators': 100}
MAE: 281.6873, RMSE: 354.7753, R²: 0.4558

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}
MAE: 266.2497, RMSE: 338.6413, R²: 0.5041

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}
MAE: 324.3596, RMSE: 391.6811, R²: 0.3367

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 5, 'learning_rate': 0.1}
MAE: 263.1106, RMSE: 328.3340, R²: 0.5339
```

Рисунок 31 – Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания CC_{50}

На данных, признаки которых были отобраны с помощью *Случайного Леса*, были получены результаты, отображенные на рисунке 32. Как можно видеть, лучше всего себя показала модель *Gradient Boosting*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 382.0124, RMSE: 442.9979, R²: 0.1515

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': None, 'min_samples_split': 3, 'n_estimators': 100}
MAE: 310.0893, RMSE: 387.4545, R²: 0.3509

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 313.2698, RMSE: 382.5921, R²: 0.3671

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 330.8017, RMSE: 406.0353, R²: 0.2871

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 300, 'l2_leaf_reg': 7, 'learning_rate': 0.05}
MAE: 315.8574, RMSE: 393.7692, R²: 0.3296
```

Рисунок 32 – Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания CC_{50}

2.2.4 Выводы

Лучше всего себя показала модель *CatBoost* на данных, признаки для которых были отобраны с помощью *Лассо*, она дала наименьшие ошибки и наивысший коэффициент детерминации – около 53% дисперсии целевой переменной. Значение коэффициента не очень высоко, но в задачах с химическими данными может быть приемлемым, с учетом, что целевая переменная плохо коррелирует с признаками.

У модели были следующие параметры, отобранные с помощью *GridSearchCV*:

- *depth*: 8
- *iterations*: 100
- *l2_leaf_reg*: 5
- *learning_rate*: 0,1

А также она показала следующие метрики:

· *MAE*: 263,1106. В среднем модель ошибается на ~263 единицы CC_{50}.

· *RMSE*: 328,3340

· R^2: 0,5339

В пределах данной задачи регрессии можно выделить несколько рекомендаций.

· Для повышения качества модели рекомендуется увеличить объём данных.

· Целесообразно привлечь специалистов в области фармакологии или хемоинформатики, чтобы интерпретировать наиболее важные признаки и проверить, действительно ли они влияют на целевой показатель.

· Для практического применения модели в реальных условиях рекомендуется протестировать её на внешних независимых выборках, отличных от обучающих данных, чтобы убедиться в обобщающей способности и воспроизводимости результатов.

2.3 Задача регрессии для SI

2.3.1 Предобработка данных

Была проведена нормализация признаков с помощью *StandardScaler*, т.к. регрессия очень чувствительна к ненормированным данным, а также на позднем этапе проведено логарифмирование с целью повышения точности предсказаний, поскольку вышеприведенные графики показывали лучшее распределение при натуральном логарифмировании. Выбросы отбираться не стали, поскольку не совсем понятно было, что можно было счесть выбросом, т.к. высокие значения индекса селективности могли указывать на неэффективный, но токсичный препарат.

2.3.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,05), с

помощью *Случайного Леса* (см. рисунок 33) и с помощью *Лассо* (см. рисунок 34).

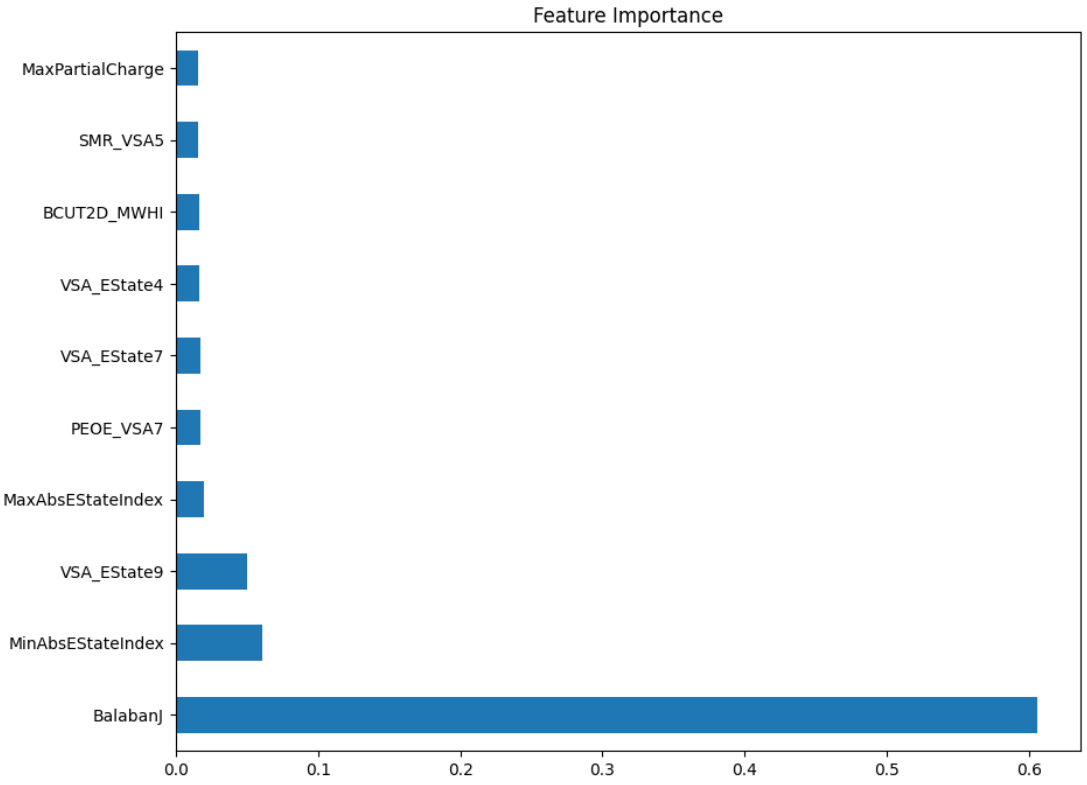

Рисунок 33 - Признаки, отобранные с помощью *Случайного Леса* для предсказания SI

```
Коэффициенты:
BalabanJ       106.115422
SMR_VSA5        66.734020
fr_NH2          62.298789
fr_Al_COO       53.888409
VSA_EState4     48.984078
                   ...
fr_methoxy     -14.933116
fr_bicyclic    -15.876431
PEOE_VSA2      -16.693194
fr_ArN         -36.186055
fr_priamide    -48.150953
Length: 94, dtype: float64
Оставлено признаков: 32
```

Рисунок 34 – Признаки, отобранные с помощью *Лассо* для предсказания SI

2.3.3 Подбор модели и гиперпараметров

На данных со всеми доступными после предобработки в разделе EDA признаками были получены результаты, отображенные на рисунке 35. Как

можно видеть, лучше всего себя показала модель, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 1.3842, RMSE: 1.7192, R²: -0.0077

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}
MAE: 1.2029, RMSE: 1.5617, R²: 0.1685

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 1.2224, RMSE: 1.5811, R²: 0.1477

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 1.2639, RMSE: 1.5988, R²: 0.1285

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.05}
MAE: 1.1835, RMSE: 1.5222, R²: 0.2101
```

Рисунок 35 – Результаты моделей на данных без отбора признаков для предсказания SI

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 36. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 1.3842, RMSE: 1.7192, R²: -0.0077

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}
MAE: 1.2029, RMSE: 1.5617, R²: 0.1685

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 1.2224, RMSE: 1.5811, R²: 0.1477

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 1.2639, RMSE: 1.5988, R²: 0.1285

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.05}
MAE: 1.1835, RMSE: 1.5222, R²: 0.2101
```

Рисунок 36 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания SI

На данных, признаки которых были отобраны с помощью *Случайного Леса*, были получены результаты, отображенные на рисунке 37. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 1.3533, RMSE: 1.7007, R²: 0.0139

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 20, 'min_samples_split': 10, 'n_estimators': 100}
MAE: 1.2950, RMSE: 1.6374, R²: 0.0859

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
MAE: 1.3359, RMSE: 1.7309, R²: -0.0215

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 1.2761, RMSE: 1.6395, R²: 0.0836

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 300, 'l2_leaf_reg': 7, 'learning_rate': 0.05}
MAE: 1.2463, RMSE: 1.5936, R²: 0.1342
```

Рисунок 37 – Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания SI

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 38. Как можно видеть, лучше всего себя показала модель *CatBoost*, дав наименьшую величину ошибки и при этом показав наивысшую метрику R^2.

```
Обучение модели: Linear Regression
Лучшие параметры: {}
MAE: 1.2668, RMSE: 1.6250, R²: 0.0997

Обучение модели: Random Forest
Лучшие параметры: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}
MAE: 1.2072, RMSE: 1.5777, R²: 0.1514

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}
MAE: 1.2518, RMSE: 1.6166, R²: 0.1090

Обучение модели: XGBoost
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}
MAE: 1.2151, RMSE: 1.6232, R²: 0.1017

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 7, 'learning_rate': 0.1}
MAE: 1.1730, RMSE: 1.5141, R²: 0.2185
```

Рисунок 38 – Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания SI

2.3.4 Выводы

Самый лучший показатель был у *CatBoost* без отбора признаков с признаками:

- *depth*: 8
- *iterations*: 300
- *l2_leaf_reg*: 7
- *learning_rate*: 0,05

Модель показала следующие метрики:

- *MAE*: 1,1745
- *RMSE*: 1,5097
- R^2: 0,2230

Поскольку наилучший результат показала модель *CatBoost* без предварительного отбора признаков, это может указывать на то, что методы

фильтрации, такие как *Лассо*, могли исключить полезные для модели переменные или внести лишний шум.

Полученные метрики продемонстрировали невысокую точность прогноза, что может быть связано с недостаточным качеством или количеством данных. Коэффициент детерминации объясняет лишь около 22% дисперсии целевой переменной.

Рекомендуется расширить обучающую выборку (т.е. увеличить датасет в целом, т.к. обучение происходило на 80% данных), привлечь профильных специалистов. Для практического же использования модели важно протестировать ее на внешних независимых наборах данных.

3 Решение задачи классификации

Задача классификации, как и задача регрессии, решается одними методами для всех целевых переменных. Ее цель состоит в построении качественной классификационной модели, которая могла бы с достаточной точностью предсказывать значения целевых.

Для того, чтобы решить задачу классификации, мы произведем несколько этапов:

1. Предобработка данных (удаление выбросов).
2. Отбор информативных признаков.
3. Логарифмирование (если необходимо) целевой переменной.
4. Подсчет медианы (если задача того требует)
5. Разделение данных на обучающую и тестовую выборки.
6. Подбор модели и гиперпараметров.
7. Сравнение моделей и выбор лучшей.

Отбор информативных признаков велся с помощью корреляции с целевой переменной, отбора *Случайным Лесом* и *Лассо*.

Для решения задач классификации были протестированы различные модели машинного обучения, отличающиеся по сложности, скорости обучения и способности к обобщению. Для каждой модели был выполнен подбор гиперпараметров с использованием *GridSearchCV*.

· *Logistic Regression*. Использовалась как базовая модель для сравнения, т.к. подходит в случае, когда между признаками и целевой переменной есть линейная связь.

· *Случайный Лес*. Это ансамблевый метод, который устойчив к шуму и переобучению и хорошо работает на небольших данных (что относится к нашему случаю).

· *Gradient Boosting* – мощная модель, которая последовательно исправляет ошибки предыдущих «деревьев». Показывает хорошие результаты в случае небольшой зашумленности данных. В данной работе использовался в чистом виде, а не для улучшения прогноза других моделей.

· *XGBoost* – одна из самых популярных моделей, по сути улучшенная реализация *градиентного бустинга*. Имеет дополнительные средства регуляризации и обладает хорошей скоростью.

· *CatBoost* – подходит для работы с категориальными данными (которые у нас присутствуют), особенно в случае, когда их много.

· *KNN (Метод k-ближайших соседей).* В данном случае это скорее базовая модель для сравнения, т.к. метод этот больше подходит для более разбитых данных, когда в пространстве они могут образовывать скопления.

Нашей задачей является задача бинарной классификации, где мы относить будем данные к классу 0 или к классу 1, которые будут что-то символизировать (например, больше ли целевой показатель определенного значения или нет), поэтому перед нами стоит вопрос сбалансированности данных. Однако, с учетом, что три задачи из четырех разбивают значения на два класса по медиане, то мы заранее можем говорить о сбалансированности данных в этих задачах.

Для оценки классификационных моделей были взяты следующие метрики:

· *Accuracy* – точность классификации. Отображает общую долю правильных предсказаний. Некорректна, если классы несбалансированы.

· *Precision* – точность положительного класса. Показывает, действительно ли выявленные элементы положительного класса относятся к этому классу или нет. Например, соединения, отнесенные к эффективным, являются таковыми?

· *Recall* – полнота положительного класса. Показывает, сколько реально элементов класса было найдено, что может указать нам на ложноположительные соединения. Например, соединения, которые модель определила, как нетоксичные, на самом деле являются очень токсичными.

· *F1-Score* – гармоническое среднее между Precision и Recall. Оно может быть полезным в случае важности обеих метрик, для нахождения

некоторого баланса между ними. Наибольшую полезность приобретает при несбалансированности данных.

· *ROC AUC* показывает насколько хорошо модель различает два класса, отражая в принципе дискриминирующую способность модели.

3.1 Задача классификации для IC_{50}

3.1.1 Предобработка данных

Как было ранее указано, распределение данных у целевой переменной IC_{50} было сильно лучше, когда ее значения были логарифмированы натуральным логарифмом. Поэтому при разделении данных на тренировочную и тестовую выборки было проведено логарифмирование. Также была проведена нормализация признаков с помощью *StandardScaler*.

Помимо этого, было проведено удаление выбросов по 3δ, как показано на рисунке 39.

Рисунок 39 – Выбросы для IC_{50}

3.1.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,1), с помощью *Случайного Леса* (см. рисунок 40) и с помощью *Лассо* (см. рисунок 41).

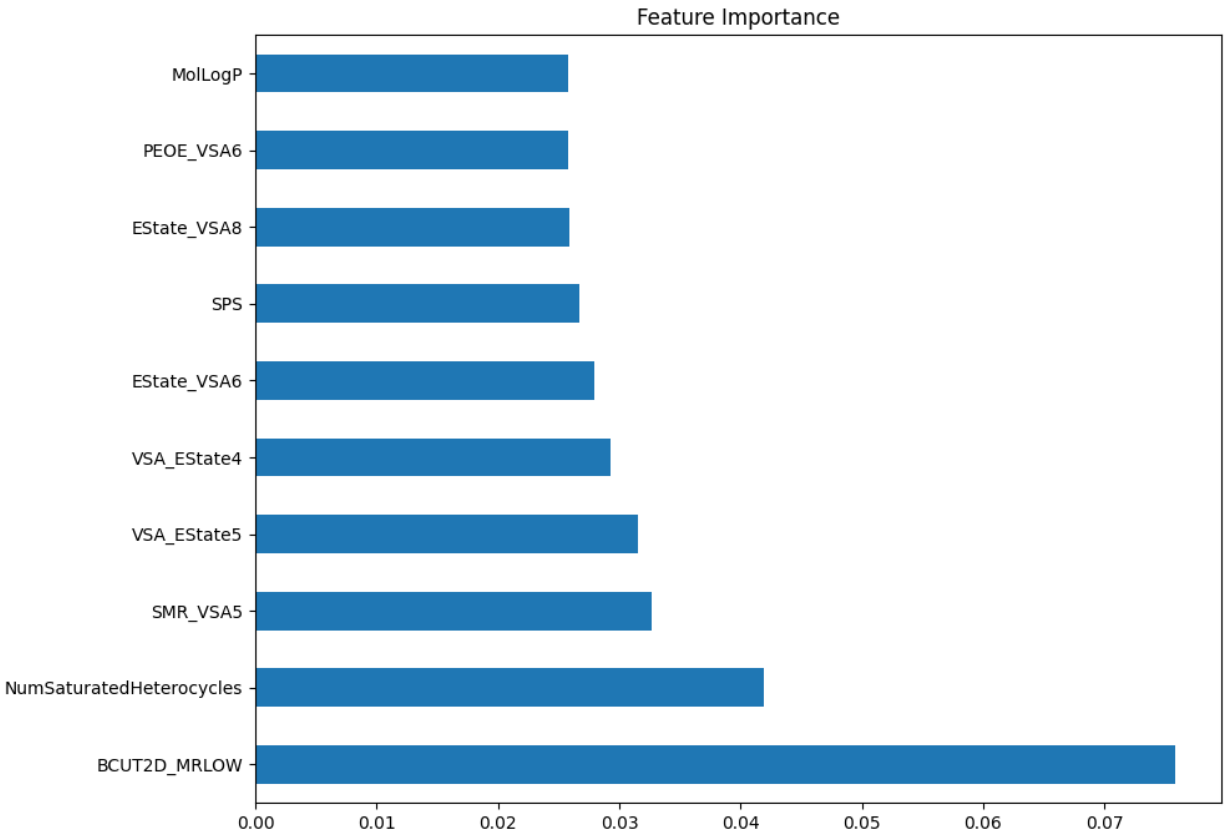

Рисунок 40 – Признаки, отобранные с помощью *Случайного Леса* для предсказания IC_{50}

```
Коэффициенты:
NumSaturatedHeterocycles    0.098755
NumAromaticHeterocycles     0.041748
BCUT2D_CHGLO                0.027276
EState_VSA7                 0.022600
fr_furan                    0.021786
                               ...
EState_VSA4                -0.035741
fr_ketone                  -0.038059
PEOE_VSA6                  -0.040101
fr_NH1                     -0.053368
fr_priamide                -0.059257
Length: 94, dtype: float64
Оставлено признаков: 35
```

Рисунок 41 – Признаки, отобранные с помощью *Лассо* для предсказания IC_{50}

3.1.3 Подбор модели и гиперпараметров

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 42. У *Случайного Леса* самые высокие *Accuracy* и *F1*. Последнее свидетельствует о хорошем балансе между *Precision* и *Recall*, хотя случай, когда мы пропустим мало эффективное лекарство нам не так критичен, как пропустить очень эффективное лекарство. Однако, и самый высокий *Precision* наблюдается у *Случайного Леса*.

У остальных моделей показатели либо средние, либо они выделяются какой-то одной метрикой, которая без других более высоких нам ничего не даст.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy: 0.6440, Precision: 0.6842, Recall: 0.6311, F1: 0.6566
ROC AUC: 0.7167

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Accuracy: 0.7068, Precision: 0.7701, Recall: 0.6505, F1: 0.7053
ROC AUC: 0.7540

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.2, 'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 100, 'subsample': 0.8}
Accuracy: 0.6387, Precision: 0.6809, Recall: 0.6214, F1: 0.6497
ROC AUC: 0.6915

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 50, 'subsample': 0.8}
Accuracy: 0.6440, Precision: 0.7273, Recall: 0.5437, F1: 0.6222
ROC AUC: 0.7359

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 7, 'learning_rate': 0.01}
Accuracy: 0.6545, Precision: 0.7342, Recall: 0.5631, F1: 0.6374
ROC AUC: 0.7492

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6545, Precision: 0.7284, Recall: 0.5728, F1: 0.6413
ROC AUC: 0.7365
```

Рисунок 42 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания IC_{50}

На данных, признаки которых были отобраны с помощью *Случайного Леса*, были получены результаты, отображенные на рисунке 43. Здесь выбор происходил между *CatBoost* и *KNN*, поскольку они оба лидировали по метрикам среди остальных моделей. У *CatBoost* наблюдались самые высокие *Precision* и *ROC AUC*, но при этом *Recall* была низковата, что означает, что у модели отличная дискриминационная способность и высокая уверенность в положительных прогнозах, но при этом и много ложных отрицаний. Если мы

можем позволить себе потерять часть эффективных лекарств, но при этом будем больше уверены в том, что те соединения, которые определились как эффективные, действительно таковыми являются, стоит выбирать эту модель.

В то же время *KNN* в данном случае является хорошо сбалансированной моделью – у нее высокие *Accuracy* и *F1*, что означает общую высокую точность модели. По совокупности метрик эта модель лучшая, и поскольку нам все же важно не пропустить как можно больше эффективных лекарств, я отдала бы предпочтение ей.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy: 0.5812, Precision: 0.6211, Recall: 0.5728, F1: 0.5960
ROC AUC: 0.6494

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200}
Accuracy: 0.6702, Precision: 0.7564, Recall: 0.5728, F1: 0.6519
ROC AUC: 0.7252

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.01, 'max_depth': 5, 'min_samples_split': 5, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.6597, Precision: 0.7500, Recall: 0.5534, F1: 0.6369
ROC AUC: 0.7166

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 1.0, 'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 100, 'subsample': 0.8}
Accuracy: 0.6649, Precision: 0.7468, Recall: 0.5728, F1: 0.6484
ROC AUC: 0.7086

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.05}
Accuracy: 0.6649, Precision: 0.7826, Recall: 0.5243, F1: 0.6279
ROC AUC: 0.7285

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 7, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6806, Precision: 0.7386, Recall: 0.6311, F1: 0.6806
ROC AUC: 0.7625
```

Рисунок 43 - Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания IC_{50}

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 44. *Логистическая регрессия* в данном случае имеет более сбалансированные метрики. У нее высокий *Precision* и неплохой *Recall*, что важно для нас. Однако, у этой модели не самая высокая точность.

Самая высокая *Accuracy* наблюдается у *KNN*, а также второй по величине *F1*, что говорит о неплохом балансе *Precision* и *Recall*, к тому же общий *ROC AUC* самый высокий. Среди всех остальных данная модель самая предпочтительная.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 100, 'penalty': 'l1', 'solver': 'liblinear'}
Accuracy: 0.6911, Precision: 0.7200, Recall: 0.6990, F1: 0.7094
ROC AUC: 0.7568

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}
Accuracy: 0.6597, Precision: 0.7111, Recall: 0.6214, F1: 0.6632
ROC AUC: 0.7534

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 7, 'min_samples_split': 5, 'n_estimators': 100, 'subsample': 0.8}
Accuracy: 0.6545, Precision: 0.6729, Recall: 0.6990, F1: 0.6857
ROC AUC: 0.7100

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 1.0, 'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.6387, Precision: 0.7024, Recall: 0.5728, F1: 0.6310
ROC AUC: 0.7198

Обучение модели: CatBoost
Лучшие параметры: {'depth': 6, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.05}
Accuracy: 0.6702, Precision: 0.7273, Recall: 0.6214, F1: 0.6702
ROC AUC: 0.7631

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 5, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.7016, Precision: 0.7674, Recall: 0.6408, F1: 0.6984
ROC AUC: 0.7644
```

Рисунок 44 - Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания IC_{50}

3.1.4 Выводы

Лучше всего себя показал *KNN*, в котором сочетался общий баланс метрик и при этом высокая уверенность в прогнозах среди всех остальных моделей. Обучена модель была на данных, признаки для которой отбирались с помощью *Лассо*.

Гиперпараметры для этой модели, которые дали наилучшие результаты, следующие:

- *n_neighbors*: 5
- *p*: 1
- *weights*: uniform

Ниже перечислены получившиеся метрики:

- *Accuracy*: 0,7016
- *Precision*: 0,7674
- *Recall*: 0,6408
- *F1*: 0,6984

· *ROC AUC*: 0,7644

Рекомендацией в данном случае может стать совет – дообучить модель на новых данных.

3.2 Задача классификации для CC_{50}

3.2.1 Предобработка данных

Как было ранее указано, распределение данных у целевой переменной CC_{50} было сильно лучше, когда ее значения были логарифмированы натуральным логарифмом. Поэтому при разделении данных на тренировочную и тестовую выборки было проведено логарифмирование. Также была проведена нормализация признаков с помощью *StandardScaler*.

Помимо этого, было проведено удаление выбросов по 3δ снизу и по 2000 сверху, как показано на рисунке 45. Верхний предел был изменен в связи с улучшением метрик моделей. Была предпринята попытка провести верхнюю черту по значению 1000, так как обсуждалось, что значения выше этого являются выбросами, однако это ухудшило предсказание моделей.

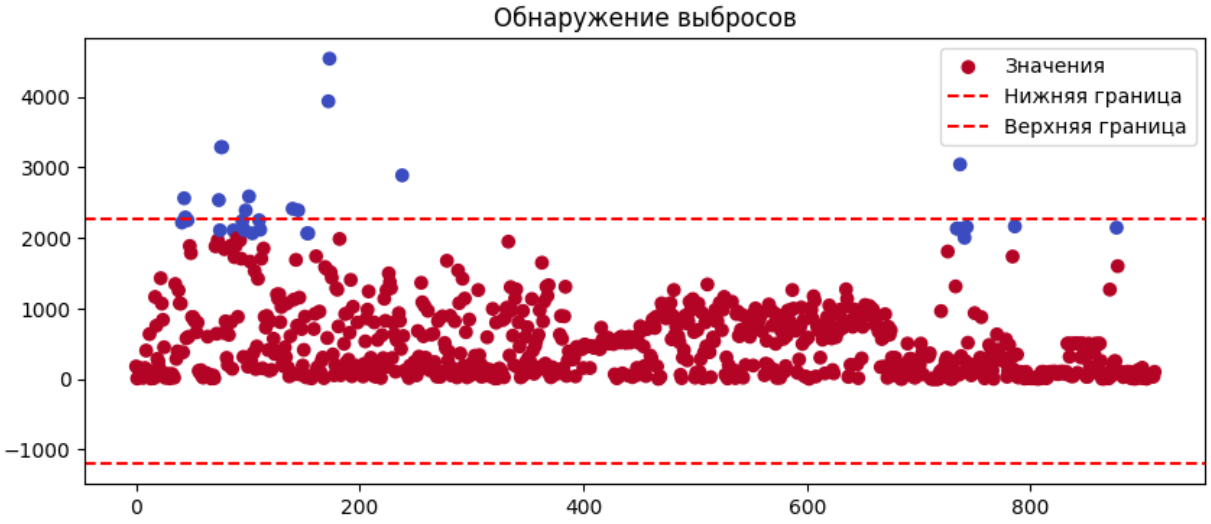

Рисунок 45 – Выбросы для CC_{50}

3.2.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,1), с

помощью *Случайного Леса* (см. рисунок 46) и с помощью *Лассо* (см. рисунок 47).

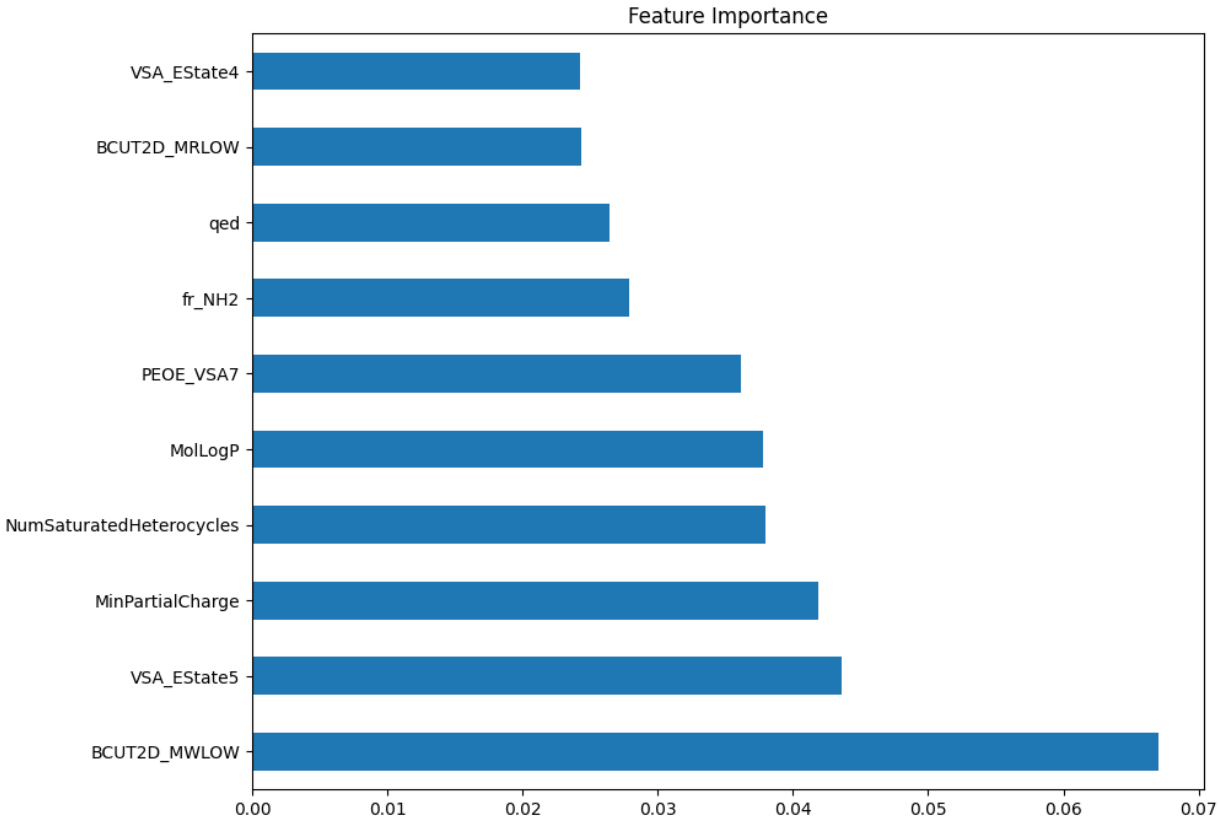

Рисунок 46 – Отбор признаков с помощью *Случайного Леса* для задачи классификации CC_{50}

```
Коэффициенты:
NumSaturatedHeterocycles    0.051759
fr_C_S                      0.025805
EState_VSA3                 0.022512
BCUT2D_CHGLO                0.018933
fr_Imine                    0.015982
                                ...
fr_sulfide                 -0.037012
PEOE_VSA7                  -0.038140
fr_allylic_oxid            -0.048405
fr_Ar_OH                   -0.068230
fr_NH2                     -0.088613
Length: 94, dtype: float64
Оставлено признаков: 22
```

Рисунок 47 – Отбор признаков с помощью *Лассо* для задачи классификации CC_{50}

3.2.3 Подбор модели и гиперпараметров

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 48.

С учетом специфики показателя, который мы пытаемся предсказать - нам важно определять верно токсичные препараты. Если исходить из того, что пропустить токсичный вариант для нас хуже, чем забраковать нетоксичный вариант, то лучше опираться на метрику *Recall*. Если же для нас критично забраковать потенциально хорошего (нетоксичного) препарата не менее, чем пропустить токсичный вариант, то стоит опираться на *F1-Score*, который основывается на *Precision* и *Recall*.

У *Логистической регрессии* самый высокий *Recall*, что означает наименьшее количество пропущенных токсичных препаратов; хороший *F1-Score*, что говорит о неплохом балансе *Precision* и *Recall*. Эта модель в рамках нашей задачи подходит больше всего.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 0.1, 'penalty': 'l1', 'solver': 'liblinear'}
Accuracy: 0.7204, Precision: 0.6916, Recall: 0.7957, F1: 0.7400
ROC AUC: 0.7940

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}
Accuracy: 0.6828, Precision: 0.7500, Recall: 0.5484, F1: 0.6335
ROC AUC: 0.7969

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_split': 2, 'n_estimators': 50, 'subsample': 1.0}
Accuracy: 0.6828, Precision: 0.7361, Recall: 0.5699, F1: 0.6424
ROC AUC: 0.8009

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 50, 'subsample': 1.0}
Accuracy: 0.6667, Precision: 0.7246, Recall: 0.5376, F1: 0.6173
ROC AUC: 0.7644

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.01}
Accuracy: 0.6989, Precision: 0.7403, Recall: 0.6129, F1: 0.6706
ROC AUC: 0.8002

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6882, Precision: 0.7011, Recall: 0.6559, F1: 0.6778
ROC AUC: 0.7894
```

Рисунок 48 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания CC_{50}

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 49. Как и у предыдущих

показателей, отдавалось предпочтение *Логистической регрессии*, поскольку она показала наивысшие результаты по *Precision*, *Recall* и *F1-Score*, что важно в рамках нашей задачи классификации.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 10, 'penalty': 'l1', 'solver': 'liblinear'}
Accuracy: 0.7796, Precision: 0.7453, Recall: 0.8495, F1: 0.7940
ROC AUC: 0.8270

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
Accuracy: 0.7419, Precision: 0.7473, Recall: 0.7312, F1: 0.7391
ROC AUC: 0.8376

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.01, 'max_depth': 7, 'min_samples_split': 2, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.7473, Precision: 0.7447, Recall: 0.7527, F1: 0.7487
ROC AUC: 0.8394

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 1.0, 'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50, 'subsample': 0.8}
Accuracy: 0.6828, Precision: 0.6809, Recall: 0.6882, F1: 0.6845
ROC AUC: 0.7504

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 7, 'learning_rate': 0.05}
Accuracy: 0.7419, Precision: 0.7368, Recall: 0.7527, F1: 0.7447
ROC AUC: 0.8488

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 10, 'p': 2, 'weights': 'uniform'}
Accuracy: 0.7366, Precision: 0.7292, Recall: 0.7527, F1: 0.7407
ROC AUC: 0.8117
```

Рисунок 49 – Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания CC_{50}

На данных, признаки которых были отобраны с помощью *Случайного Леса*, были получены результаты, отображенные на рисунке 50. У модели *KNN* наблюдаются наивысшие *Accuracy*, *Recall* и *ROC AUC*, что говорит о наименьшем количестве пропущенных токсичных соединений (среди остальных моделей), хорошей дискриминационной способности. Эта модель выделяется сбалансированностью метрик.

У *Логистической регрессии* хороший баланс между *Precision* и *Recall*, и последний – самый высокий среди остальных моделей, как и F1-Score. С учетом выше изложенных требований к модели данной задачи, предпочтительнее будет выбрать *Логистическую регрессию*.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy: 0.6882, Precision: 0.6842, Recall: 0.6989, F1: 0.6915
ROC AUC: 0.7173

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}
Accuracy: 0.6452, Precision: 0.7288, Recall: 0.4624, F1: 0.5658
ROC AUC: 0.7656

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_split': 5, 'n_estimators': 50, 'subsample': 0.8}
Accuracy: 0.6667, Precision: 0.7460, Recall: 0.5054, F1: 0.6026
ROC AUC: 0.7698

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50, 'subsample': 0.8}
Accuracy: 0.5968, Precision: 0.6250, Recall: 0.4839, F1: 0.5455
ROC AUC: 0.7033

Обучение модели: CatBoost
Лучшие параметры: {'depth': 6, 'iterations': 100, 'l2_leaf_reg': 1, 'learning_rate': 0.01}
Accuracy: 0.6452, Precision: 0.7143, Recall: 0.4839, F1: 0.5769
ROC AUC: 0.7492

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6989, Precision: 0.7033, Recall: 0.6882, F1: 0.6957
ROC AUC: 0.7837
```

Рисунок 50 – Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания CC_{50}

3.2.4 Выводы

Среди всех представленных моделей наилучшие показатели были у *Логистической регрессии*, обученной на данных, признаки у которых отбирались с помощью *Лассо*. Сочетание высокого *Recall* и чуть менее высокого *Precision*, с хорошей оценкой *F1-Score* баланса первых двух метрик делает эту модель наиболее надежным вариантом среди остальных.

Гиперпараметры для данной модели следующие:

- *C*: 10
- *penalty*: l1
- *solver*: liblinear

Ее метрики были:

- *Accuracy*: 0.7796
- *Precision*: 0.7453
- *Recall*: 0.8495
- *F1*: 0.7940
- *ROC AUC*: 0.8270

3.3 Задача классификации для SI (разделение по медиане)

3.3.1 Предобработка данных

Была проведена нормализация признаков с помощью *StandardScaler*, а также на позднем этапе проведено логарифмирование с целью повышения точности предсказаний, поскольку вышеприведенные графики показывали лучшее распределение при натуральном логарифмировании. Выбросы отбираться не стали, поскольку не совсем понятно было, что можно было счесть выбросом, т.к. высокие значения индекса селективности могли указывать на неэффективный, но токсичный препарат.

3.3.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,1), с помощью *Случайного Леса* (см. рисунок 51) и с помощью *Лассо* (см. рисунок 52).

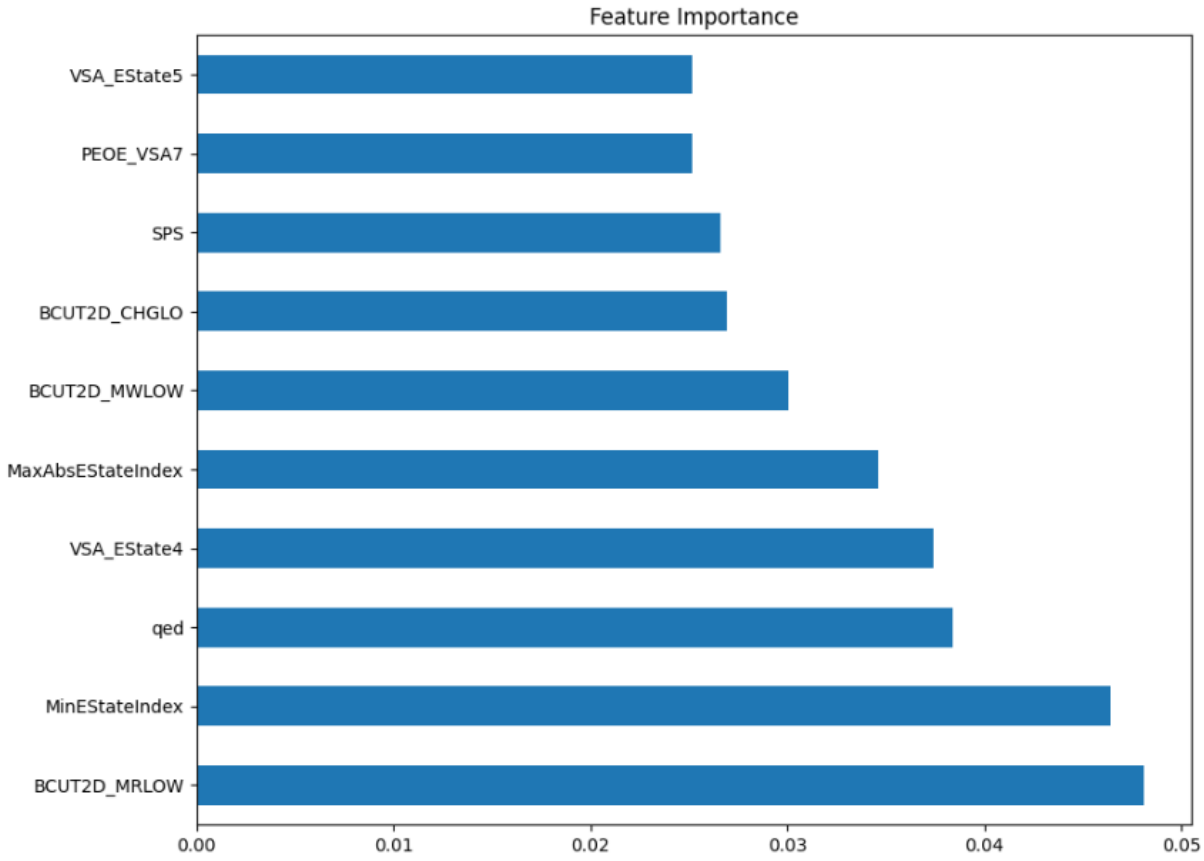

Рисунок 51 - Признаки, отобранные с помощью *Случайного Леса* для предсказания SI

```
Коэффициенты:
fr_NH2                      0.020193
fr_Imine                    0.017284
MinEStateIndex              0.005634
VSA_EState5                 0.002011
SPS                         0.000295
                               ...
MaxPartialCharge           -0.003266
fr_methoxy                 -0.004464
fr_nitro                   -0.007485
PEOE_VSA14                 -0.016419
NumSaturatedHeterocycles   -0.046558
Length: 94, dtype: float64
Оставлено признаков: 14
```

Рисунок 52 – Признаки, отобранные с помощью *Лассо* для предсказания SI

3.3.3 Подбор модели и гиперпараметров

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 53. У *KNN* наблюдаются наивысшие *Recall* и *F1*, что говорит о неплохой точности модели. Хоть у *CatBoost* более высокий *ROC AUC*, предпочтение все же отдаем *KNN*, поскольку нам важно не пропустить высокие индексы селективности, которые свидетельствуют обычно о токсичных препаратах.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy: 0.6186, Precision: 0.5745, Recall: 0.6136, F1: 0.5934
ROC AUC: 0.6531

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 100}
Accuracy: 0.6289, Precision: 0.5870, Recall: 0.6136, F1: 0.6000
ROC AUC: 0.6786

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.01, 'max_depth': 3, 'min_samples_split': 2, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.6289, Precision: 0.5909, Recall: 0.5909, F1: 0.5909
ROC AUC: 0.6744

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 1.0, 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100, 'subsample': 1.0}
Accuracy: 0.6186, Precision: 0.5761, Recall: 0.6023, F1: 0.5889
ROC AUC: 0.6469

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 200, 'l2_leaf_reg': 1, 'learning_rate': 0.01}
Accuracy: 0.6443, Precision: 0.6067, Recall: 0.6136, F1: 0.6102
ROC AUC: 0.7009

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 10, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6495, Precision: 0.6000, Recall: 0.6818, F1: 0.6383
ROC AUC: 0.6756
```

Рисунок 53 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания SI

На данных, признаки которых были отобраны с помощью Случайного Леса, были получены результаты, отображенные на рисунке 54. Модель *KNN* продемонстрировала лучшие метрики по *полноте* и *F1*, к тому же ее *точность* не настолько сильно отстает от других моделей, как и оценка ее дискриминационной способности.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy: 0.5412, Precision: 0.4955, Recall: 0.6250, F1: 0.5528
ROC AUC: 0.5794

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 50}
Accuracy: 0.6443, Precision: 0.6092, Recall: 0.6023, F1: 0.6057
ROC AUC: 0.6976

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.2, 'max_depth': 5, 'min_samples_split': 5, 'n_estimators': 50, 'subsample': 1.0}
Accuracy: 0.6289, Precision: 0.5870, Recall: 0.6136, F1: 0.6000
ROC AUC: 0.6609

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100, 'subsample': 1.0}
Accuracy: 0.5773, Precision: 0.5312, Recall: 0.5795, F1: 0.5543
ROC AUC: 0.6354

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 200, 'l2_leaf_reg': 3, 'learning_rate': 0.1}
Accuracy: 0.6701, Precision: 0.6429, Recall: 0.6136, F1: 0.6279
ROC AUC: 0.7011

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 7, 'p': 1, 'weights': 'distance'}
Accuracy: 0.6495, Precision: 0.5926, Recall: 0.7273, F1: 0.6531
ROC AUC: 0.6972
```

Рисунок 54 – Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания SI

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 55. У *XGBoost* в данном случае наивысшие метрики по *Recall* и *F1*, что для нас в приоритете. К тому же по другим метрикам модель уступает остальным не сильно значительно, чтобы мы могли предпочесть что-то другое.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 10, 'penalty': 'l1', 'solver': 'liblinear'}
Accuracy: 0.6546, Precision: 0.6129, Recall: 0.6477, F1: 0.6298
ROC AUC: 0.6678

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
Accuracy: 0.6082, Precision: 0.5667, Recall: 0.5795, F1: 0.5730
ROC AUC: 0.6616

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.2, 'max_depth': 7, 'min_samples_split': 5, 'n_estimators': 100, 'subsample': 1.0}
Accuracy: 0.5515, Precision: 0.5051, Recall: 0.5682, F1: 0.5348
ROC AUC: 0.6162

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 50, 'subsample': 0.8}
Accuracy: 0.6340, Precision: 0.5825, Recall: 0.6818, F1: 0.6283
ROC AUC: 0.6392

Обучение модели: CatBoost
Лучшие параметры: {'depth': 4, 'iterations': 200, 'l2_leaf_reg': 3, 'learning_rate': 0.1}
Accuracy: 0.6186, Precision: 0.5761, Recall: 0.6023, F1: 0.5889
ROC AUC: 0.6758

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 10, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6134, Precision: 0.5765, Recall: 0.5568, F1: 0.5665
ROC AUC: 0.6833
```

Рисунок 55 – Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания SI

3.3.4 Выводы

Лучше всего себя показала модель *KNN*, обученная на признаках, отобранных через *Случайный Лес*. У нее меньше всего пропущенных потенциально токсичных препаратов, в отличие от других, а также хороший баланс между *Precision* и *Recall*. Эта модель также отличилась хорошей дискриминационной способностью.

Ее гиперпараметры:

- *n_neighbors*: 7
- *p*: 1
- *weights*: distance

Ее метрики:

- *Accuracy*: 0.6495
- *Precision*: 0.5926
- *Recall*: 0.7273
- *F1*: 0.6531
- *ROC AUC*: 0.6972

3.4 Задача классификации для SI (разделение по значению 8)

3.4.1 Предобработка данных

Была проведена нормализация признаков с помощью *StandardScaler*, а также на позднем этапе проведено логарифмирование с целью повышения точности предсказаний, поскольку вышеприведенные графики показывали лучшее распределение при натуральном логарифмировании. Прологарифмировано было и значение 8, чтобы не потерять верную классификацию. Выбросы отбираться не стали, поскольку не совсем понятно было, что можно было счесть выбросом, т.к. высокие значения индекса селективности могли указывать на неэффективный, но токсичный препарат.

3.4.2 Отбор признаков

Признаки отбирались несколькими способами: по корреляции с целевой переменной (отбирались признаки, корреляция которых начиналась от 0,1), с помощью *Случайного Леса* (см. рисунок 56) и с помощью *Лассо* (см. рисунок 57).

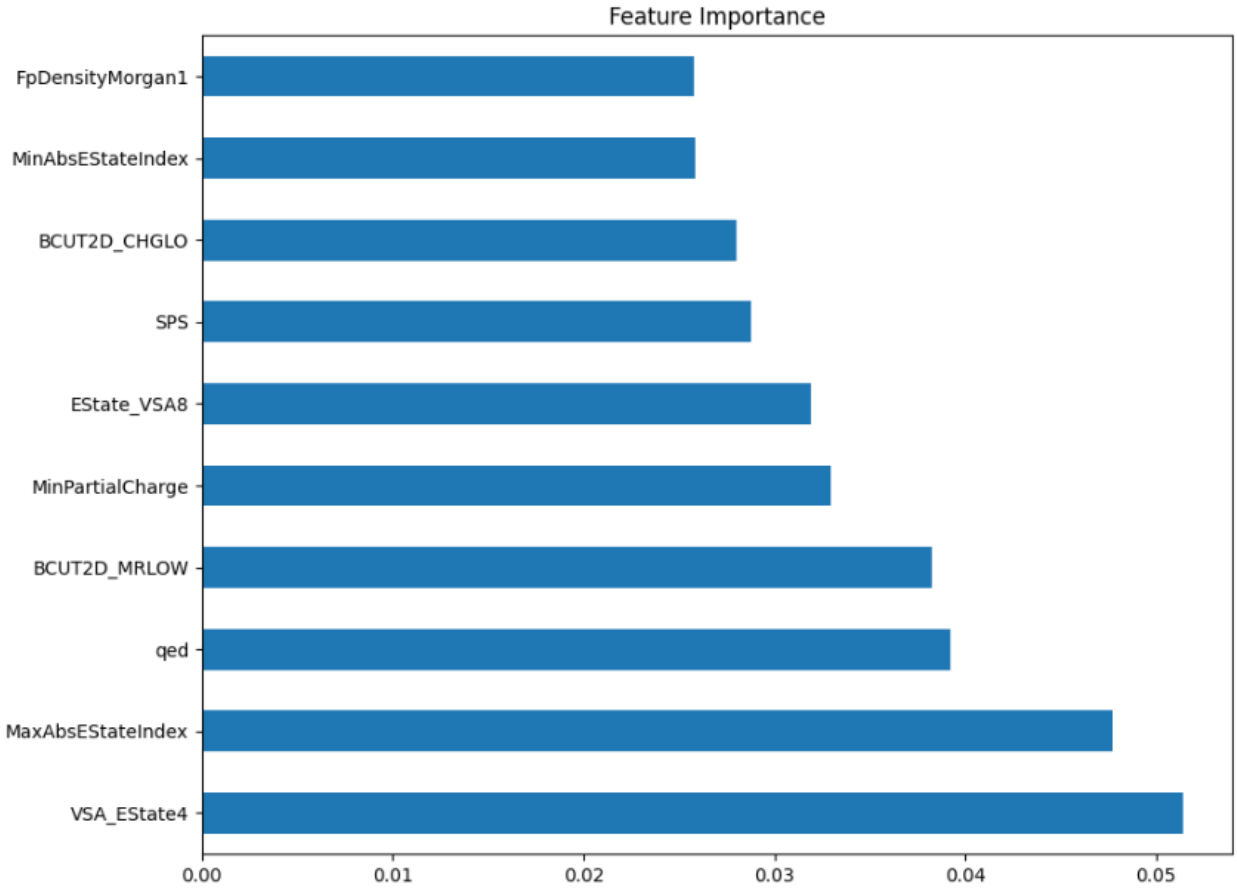

Рисунок 56 - Признаки, отобранные с помощью *Случайного Леса* для предсказания SI

```
Коэффициенты:
fr_Imine                 0.065254
SMR_VSA6                 0.047159
fr_NH1                   0.044132
FpDensityMorgan1         0.041673
fr_priamide              0.039030
                             ...
SlogP_VSA3              -0.027342
fr_nitro                -0.027612
MinPartialCharge        -0.040936
fr_allylic_oxid         -0.057654
NumSaturatedHeterocycles -0.080568
Length: 94, dtype: float64
Оставлено признаков: 46
```

Рисунок 57 – Признаки, отобранные с помощью *Лассо* для предсказания SI

3.4.3 Подбор модели и гиперпараметров

На данных, признаки которых были отобраны с помощью корреляции, были получены результаты, отображенные на рисунке 58. Лучше всего себя

показала модель *CatBoost*, поскольку у нее самый высокий *F1-Score*, а также вторая лучшая метрика *Recall*, к тому же модель продемонстрировала самую лучшую дискриминационную способность (*ROC AUC*).

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 10, 'penalty': 'l1', 'solver': 'liblinear'}
Accuracy: 0.6392, Precision: 0.4921, Recall: 0.4493, F1: 0.4697
ROC AUC: 0.6110

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
Accuracy: 0.6701, Precision: 0.5581, Recall: 0.3478, F1: 0.4286
ROC AUC: 0.6437

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_split': 5, 'n_estimators': 50, 'subsample': 0.8}
Accuracy: 0.6649, Precision: 0.5500, Recall: 0.3188, F1: 0.4037
ROC AUC: 0.6391

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100, 'subsample': 1.0}
Accuracy: 0.6443, Precision: 0.5000, Recall: 0.3913, F1: 0.4390
ROC AUC: 0.6375

Обучение модели: CatBoost
Лучшие параметры: {'depth': 6, 'iterations': 200, 'l2_leaf_reg': 7, 'learning_rate': 0.05}
Accuracy: 0.6649, Precision: 0.5357, Recall: 0.4348, F1: 0.4800
ROC AUC: 0.6792

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 10, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6907, Precision: 0.6047, Recall: 0.3768, F1: 0.4643
ROC AUC: 0.6716
```

Рисунок 58 – Результаты моделей на данных, признаки которых отобраны с помощью корреляции для предсказания SI

На данных, признаки которых были отобраны с помощью *Случайного Леса*, были получены результаты, отображенные на рисунке 59. У *Случайного Леса* наилучшие *Accuracy*, *Precision*, отличный *F1* и умеренный *Recall*. Эта модель среди остальных даст большую стабильность и уверенность в прогнозах.

```
Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100}
Accuracy: 0.7113, Precision: 0.6512, Recall: 0.4058, F1: 0.5000
ROC AUC: 0.6825

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.01, 'max_depth': 3, 'min_samples_split': 5, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.6546, Precision: 0.5385, Recall: 0.2029, F1: 0.2947
ROC AUC: 0.6644

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.6701, Precision: 0.6190, Recall: 0.1884, F1: 0.2889
ROC AUC: 0.6823

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 100, 'l2_leaf_reg': 3, 'learning_rate': 0.05}
Accuracy: 0.7010, Precision: 0.6341, Recall: 0.3768, F1: 0.4727
ROC AUC: 0.6631

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 10, 'p': 1, 'weights': 'uniform'}
Accuracy: 0.6598, Precision: 0.5294, Recall: 0.3913, F1: 0.4500
ROC AUC: 0.6624
```

Рисунок 59 – Результаты моделей на данных, признаки которых отобраны с помощью *Случайного Леса* для предсказания SI

На данных, признаки которых были отобраны с помощью *Лассо*, были получены результаты, отображенные на рисунке 60. Модель *KNN* продемонстрировала наивысшие метрики *Recall*, *F1*, что для нас приоритетнее по причинам, упоминавшихся выше.

```
Обучение модели: Logistic Regression
Лучшие параметры: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy: 0.6907, Precision: 0.5763, Recall: 0.4928, F1: 0.5312
ROC AUC: 0.7184

Обучение модели: Random Forest
Лучшие параметры: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}
Accuracy: 0.6701, Precision: 0.5758, Recall: 0.2754, F1: 0.3725
ROC AUC: 0.6941

Обучение модели: Gradient Boosting
Лучшие параметры: {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_split': 2, 'n_estimators': 50, 'subsample': 1.0}
Accuracy: 0.6701, Precision: 0.5556, Recall: 0.3623, F1: 0.4386
ROC AUC: 0.6958

Обучение модели: XGBoost
Лучшие параметры: {'colsample_bytree': 0.7, 'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 200, 'subsample': 0.8}
Accuracy: 0.6804, Precision: 0.6296, Recall: 0.2464, F1: 0.3542
ROC AUC: 0.7014

Обучение модели: CatBoost
Лучшие параметры: {'depth': 8, 'iterations': 200, 'l2_leaf_reg': 1, 'learning_rate': 0.01}
Accuracy: 0.7062, Precision: 0.6364, Recall: 0.4058, F1: 0.4956
ROC AUC: 0.6954

Обучение модели: KNN
Лучшие параметры: {'n_neighbors': 5, 'p': 1, 'weights': 'distance'}
Accuracy: 0.6443, Precision: 0.5000, Recall: 0.5217, F1: 0.5106
ROC AUC: 0.6547
```

Рисунок 60 – Результаты моделей на данных, признаки которых отобраны с помощью *Лассо* для предсказания SI

3.4.4 Выводы

Лучше всего себя показала модель *KNN*, обученная на признаках, отобранных при помощи Лассо. Она пропускает меньше препаратов с высоким индексом селективности в отличие от других, и у нее лучший баланс между Precision и Recall с приемлемой точностью.

Ее гиперпараметры:

- *n_neighbors*: 5
- *p*: 1
- *weights*: distance

Ее метрики:

- *Accuracy*: 0,6443
- *Precision*: 0,5000
- *Recall*: 0,5217
- *F1*: 0,5106
- *ROC AUC*: 0,6547

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были проведены всесторонний анализ данных, построение моделей регрессии и классификации для прогнозирования ключевых показателей эффективности и токсичности лекарственных соединений: IC_{50}, CC_{50} и SI (индекс селективности).

На этапе разведочного анализа были удалены дубликаты и пропущенные значения. Было выполнено описание признаков и целевых переменных и установлено, что корреляция между целевыми переменными и между признаками была низкой, что указывает на сложность задачи прогнозирования. Для устранения мультиколлинеарности были удалены по одному из пары сильно коррелирующих признаков. Этот этап позволил подготовить данные к дальнейшему моделированию, повысив их качество и уменьшив риск переобучения.

Наиболее успешной моделью в задачах регрессии себя показала *CatBoost*, на предсказании всех трех целевых переменных дав наивысшие метрики среди всех остальных моделей. Показатели, впрочем, были не столь удовлетворительны, причина чего могла скрываться в недостатке данных. Для повышения точности рекомендовалось увеличить объем данных, расширить набор признаков и привлечь специалистов для экспертной интерпретации важных показателей.

Наиболее успешными моделями в задачах классификации стали *KNN* и *Логистическая регрессия*. Хороший баланс метрик и уверенность в прогнозах делала *KNN* подходящей для использования, и в перспективе ее качество можно было бы улучшить дообучением на новых данных. А *Логистическая регрессия* обеспечивала самый высокий *Recall*, что особенно было важно в задаче выявления потенциально опасных препаратов. Она демонстрировала хороший баланс между *точностью* и *полнотой*, что делало ее надежным решением.

Основными рекомендациями для улучшения результатов стали советы по увеличению объема тренировочных данных, расширению набора

признаков, проверки моделей на дополнительных тестовых данных, а также по привлечению специалистов из профильных областей для выявления важных факторов, которые могут потенциально влиять на прогноз и классификацию.