Национальный исследовательский ядерный университет «МИФИ»

Классическое машинное обучение

Курсовая работа

«Активность соединений лекарственных препаратов»

Студент:

Смородин Борис Борисович

Содержание

Введение

Разработка новых лекарственных препаратов начинается с идентификации перспективных молекул, обладающих высокой биологической активностью и минимальной токсичностью. На ранних этапах исследования дорогостоящие экспериментальные методы (in vitro и in vivo) заменяются вычислительными подходами, которые позволяют эффективно отбирать кандидаты для последующего тестирования. В этом контексте ключевую роль играют методы химоинформатики и машинного обучения, обеспечивающие прогнозирование фармакологических характеристик соединений на основе их молекулярной структуры и физико-химических свойств.

В рамках стандартного поля задач классического машинного обучения могут быть решены следующие задачи:

1. Количественный прогноз параметров активности (IC_{50}), токсичности (CC_{50}) и селективности (SI), что помогает оценить эффективность и безопасность соединений.

2. Классификация молекул по категориям (например, «сильные/слабые ингибиторы», «токсичные/нетоксичные»), что упрощает отбор перспективных кандидатов.

В рамках данной курсовой работы исследуется набор данных, содержащий информацию о 1001 химическом соединении, описанных через 214 структурных, электронных и топологических дескрипторов. Построенные модели машинного обучения для удовлетворения цели курсовой работы.

Цель курсовой работы

На основе анализа физико-химических и структурных характеристик химических соединений необходимо построить и сравнить модели машинного обучения для прогнозирования ключевых показателей биологической

активности (IC_{50}, CC_{50}, SI) и классификации соединений на категории (сильные/слабые ингибиторы, токсичные/нетоксичные и т.д.). Результаты должны обеспечить возможность ранжирования веществ по эффективности и безопасности, а также выявить структурные дескрипторы, влияющие на активность и селективность, что позволит подбирать оптимальные составы лекарственных препаратов.

Задачи

1. Предобработка данных:

- Провести исследовательский анализ данных (EDA), оценить информативность признаков.

2. Построить регрессионные модели машинного обучения для прогнозирования количественных значений:

- IC_{50} (активность)
- CC_{50} (токсичность)
- SI (селективность)

3. Построить модели классификации для прогнозирования:

- IC_{50} больше медианного значения IC_{50}
- CC_{50} больше медианного значения CC_{50}
- SI больше медианного значения SI
- SI больше 8

4. Сравнить эффективность полученных моделей в рамках каждой задачи по выбранным метрикам.

5. Оценить влияние признаков на результаты работы модели.

6. Исследовать возможность повышения качества работы модели.

7. Провести сравнительный анализ и оптимизацию моделей МО.

8. Провести интерпретацию полученных результатов и сформировать предложения по использованию моделей.

1 Разведывательный анализ данных (EDA)

Для успешного решения задач прогнозирования биологической активности химических соединений ключевое значение имеет качественная предобработка данных и глубокий разведывательный анализ (EDA).

В рамках данного этапа исследуются статистические свойства признаков, выявляются корреляции между переменными, анализируются распределения целевых показателей (IC_{50}, CC_{50}, SI) и их зависимость от структурных дескрипторов.

Особое внимание уделяется обработке выбросов, устранению линейной зависимости, мультиколлинеарности и преобразованию признаков (а также целевой величины) для повышения устойчивости и качества работы моделей.

Результаты EDA не только обеспечивают корректную подготовку данных для машинного обучения, но и позволяют выявить ключевые факторы, влияющие на эффективность и безопасность соединений.

1.1 Аналитика данных

В рамках аналитической части EDA были проведены работы по первичному и более глубокому исследованию набора данных. Также были сформированы гипотезы о влиянии преобразований признаков и целевых переменных на результаты работы модели. Было получено приблизительное понимание того, каким образом будут решаться дальнейшие поставленные задачи.

1.1.1 Описание данных

Как было указано во введении, набор данных представляет собой табличную информацию, описывающую 1001 химическое соединение при помощи 214 числовых признаков.

1. Общие молекулярные дескрипторы

- MolWt — молекулярная масса.
- HeavyAtomCount — количество тяжёлых атомов (без H).
- NumValenceElectrons — валентные электроны.
- NumRadicalElectrons — неспаренные электроны.
- FractionCSP3 — доля sp^3-гибридизованных атомов C.
- TPSA — топологическая полярная поверхность (проницаемость через мембраны).
- LabuteASA — доступная поверхность по Labute (взаимодействие с растворителем).
- QED — оценка «лекарственности» (комплексный показатель).
- MolLogP — гидрофобность (logP).
- MolMR — молекулярная рефрактивность (поляризуемость).

Примечание: Дескриптор SPS (сложность синтеза) исключён, так как не влияет на задачу.

2. Электронные дескрипторы

- Max/MinPartialCharge — экстремальные значения частичных зарядов.
- PEOE_VSA — распределение зарядов (метод PEOE).
- EState_VSA — зарядовое состояние + топология.
- Max/MinEStateIndex — индексы электротопологического состояния.

3. Топологические дескрипторы

- Chi0-Chi4v — индексы связности (топология, разветвление).
- Kappa1-Kappa3 — индексы формы и компактности.
- HallKierAlpha — стерическая насыщенность.
- BalabanJ — связность и цикличность (разветвлённость).
- Ipc, AvgIpc, BertzCT — информационные индексы сложности структуры.

4. BCUT-дескрипторы

- BCUT2D_MW — молекулярная масса (высокая/низкая).
- BCUT2D_CHG — заряд (высокий/низкий).
- BCUT2D_LOGP — гидрофобность (высокая/низкая).
- BCUT2D_MR — рефрактивность (высокая/низкая).

5. VSA-дескрипторы

- SMR_VSA1–10 — молекулярная рефрактивность по диапазонам.
- SlogP_VSA1–12 — гидрофобность по участкам.
- EState_VSA1–10 — электротопология по поверхности.
- PEOE_VSA1–14 — частичные заряды по диапазонам.

6. Отпечатки (Morgan fingerprints)

FpDensityMorgan1–3 — плотность структурных фрагментов (радиусы 1, 2, 3).

7. Фрагментные дескрипторы

- Фенолы: fr_phenol, fr_Ar_OH.
- Амины: fr_NH2, fr_aniline.
- Азосоединения: fr_azide, fr_azo.
- Галогены: fr_halogen, fr_alkyl_halide.
- Барбитураты: fr_barbitur.
- Нитро-соединения: fr_nitro, fr_nitro_arom.
- Кольца: fr_benzene, fr_pyridine, fr_furan.

8. Структурные количественные дескрипторы

- NumHAcceptors/NumHDonors — акцепторы/доноры водородных связей.
- NumRotatableBonds — вращающиеся связи.
- NumAromatic/Aliphatic/SaturatedRings — типы колец.
- NumHeteroatoms — количество гетероатомов.

RingCount — общее число колец.

1.1.2 Первичный анализ данных

В рамках первичного (базового) анализа данных были проделаны работы, позволяющие ознакомиться с датасетом и устранить его базовые недостатки, такие как:

- Наличие пустых данных;
- Наличие дубликатов;

Были обнаружены 3 строки с пропусками, названия признаков, в которых содержались пропуски указаны на рисунке 1.

```
Колонки с пропусками:
MaxPartialCharge       3
MinPartialCharge       3
MaxAbsPartialCharge    3
MinAbsPartialCharge    3
BCUT2D_MWHI            3
BCUT2D_MWLOW           3
BCUT2D_CHGHI           3
BCUT2D_CHGLO           3
BCUT2D_LOGPHI          3
BCUT2D_LOGPLOW         3
BCUT2D_MRHI            3
BCUT2D_MRLOW           3
dtype: int64
```

Рисунок 1 – названия признаков, содержащих пропуски

Поскольку строк с пустыми значениями мало (около 0.3% от размера набора данных) в целях не ухудшения достоверности, а также ввиду комплексности данных, было принято решение удалить строки, содержащие пропущенные значения.

В результате в наборе данных осталось 998 строк.

В рамках проведения данного этапа работ было обнаружено 32 дубликата. Наличие дубликатов в выборке приводит к тому, что модель может:

1. Переобучиться, поскольку будет встречать одни и те же закономерности чаще других;
2. Показать недостоверные результаты при испытаниях на тестовой выборке, поскольку как минимум часть дубликатов может попасть как в обучающую, так и тестовую выборки.

По этим причинам было принято решения удалить дублирующиеся строки.

После их удаления в датасете осталось 966 строк.

1.1.3 Анализ распределения целевых переменных

Первичные графики распределения целевых переменных приведены на рисунках 2 – 4.

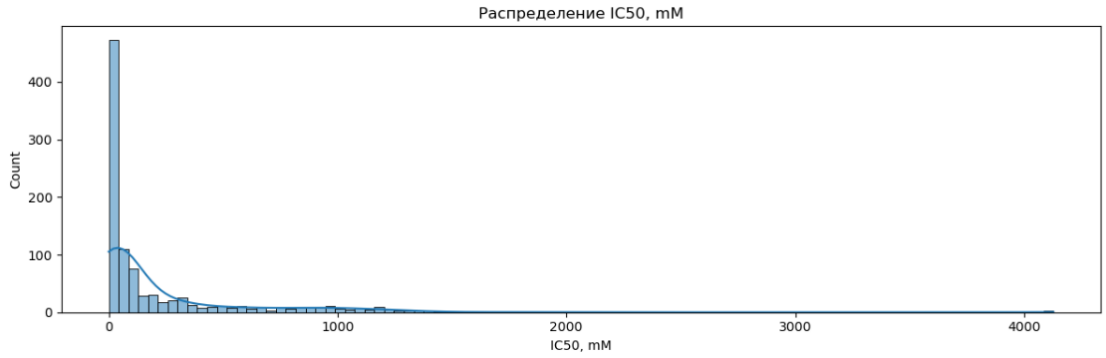

Рисунок 2 – распределение IC_{50}

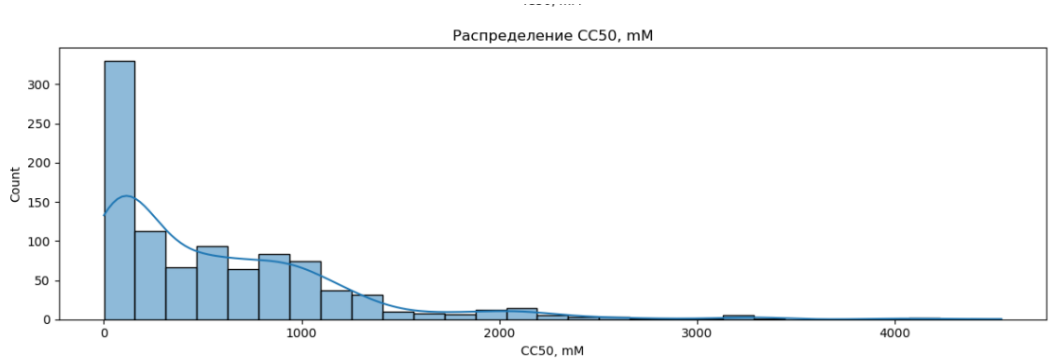

Рисунок 3 – распределение CC_{50}

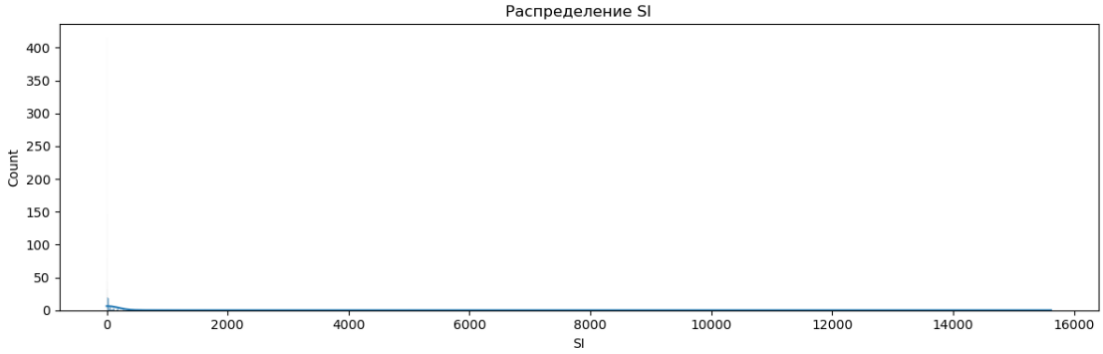

Рисунок 4 – распределение SI

Для целевых переменных IC_{50} и CC_{50} наблюдаются левоассиметричные распределения с явно выраженным правым хвостом, свидетельствующем об обилии выбросов.

Распределение этих целевых переменных имеет сходства с логарифмическим распредлением: все значения больше нуля, при этом наблюдается левосторонняя ассиметрия с правым хвостом. Следовательно, логарифмирование может помочь улучшить картину распределения. Распределения IC_{50} и CC_{50} после логарифмирования представлены на рисунках 5 – 6. Был использован десятичный логарифм.

Распределение SI выглядит менее однозначно: общая масса значений сконцентрирована около нуля. Для того, чтобы визуализировать распределение более наглядно, был построен scatterplot (рисунок 7), который будет использован в том числе для отсеивания выбросов.

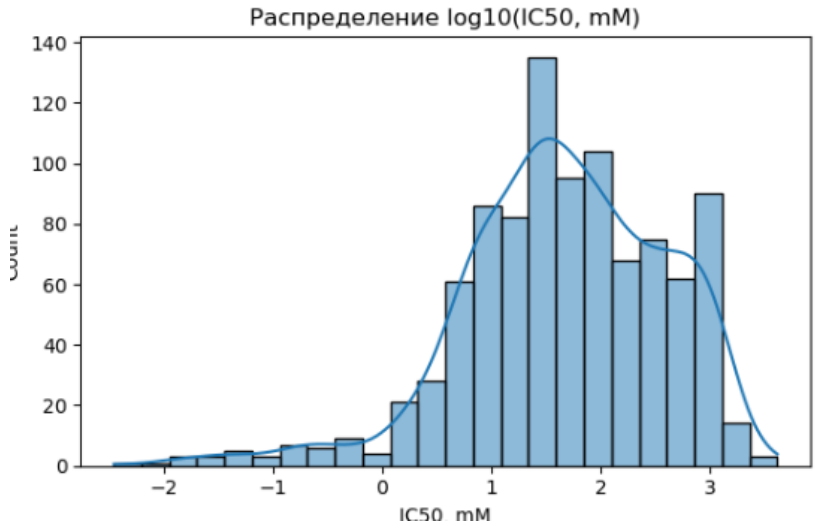

Рисунок 5 – распределение прологарифмированного IC_{50}

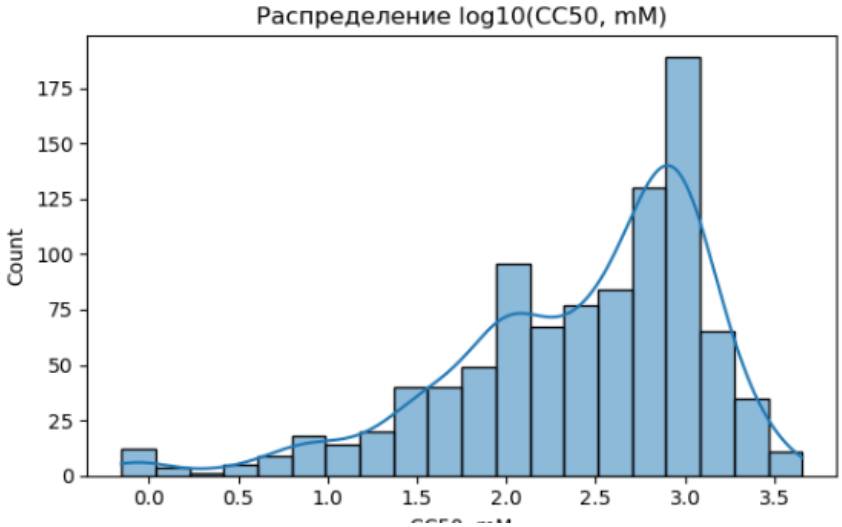

Рисунок 6 – распределение прологарифмированного CC_{50}

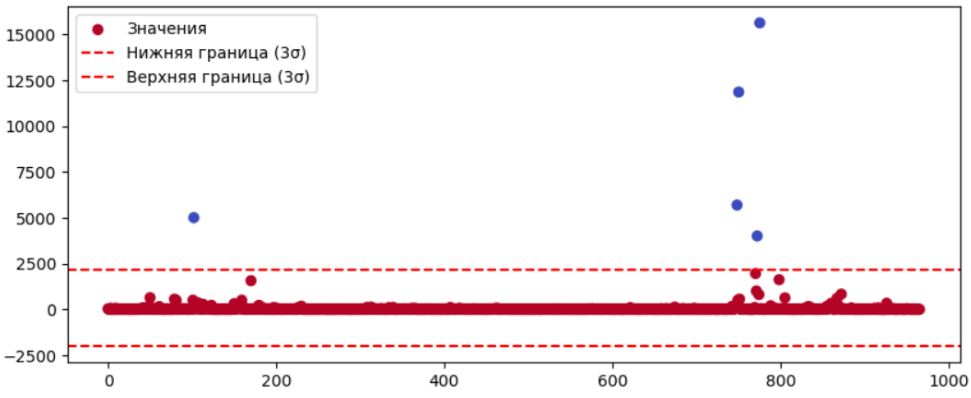

Рисунок 7 – распределение для SI по точкам

Из рисунков 5 – 6 очевидно, что логарифмирование помогает улучшить распределение целевых признаков IC_{50} и SI_{50}. Это может положительно

сказаться на результатах решения регрессионной задачи, позволяя использовать модели, чувствительные к форме распределения, например, линейные.

Из рисунка 7 очевидно, что целевая величина SI имеет подавляющее значение точек в области, находящейся крайне близко к нулю. Логарифмирование в данном случае не является целесообразным, достаточно будет отсеять выбросы.

1.1.4 Корреляционный анализ целевых и нецелевых признаков

Цель корреляционного анализа – определить корреляцию признаков (целевых и нецелевых) между собой для того, чтобы определить решающий набор признаков при обучении моделей.

Определим корреляцию целевых признаков друг между другом, построив матрицу корреляции (рисунок 8).

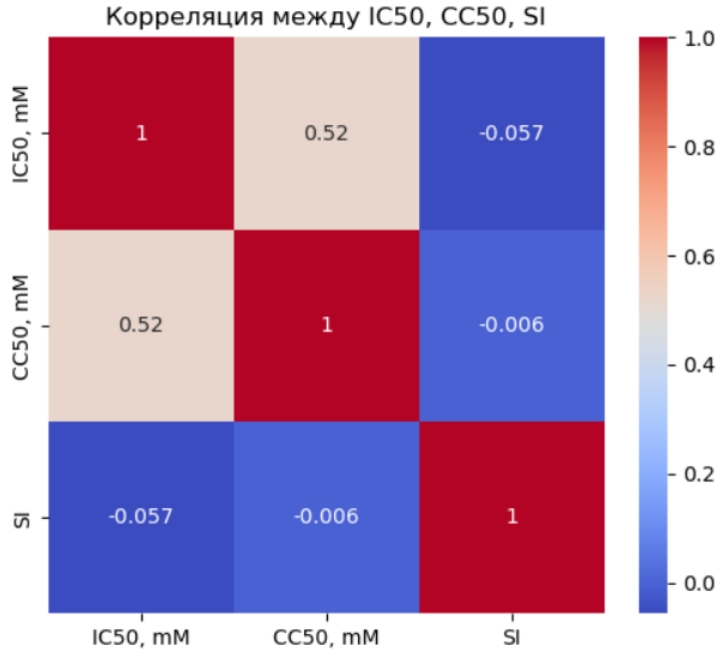

Рисунок 8 – корреляция между целевыми величинами

Как видно из матрицы корреляции, наблюдается средняя корреляция между IC_{50} и CC_{50}, что указывает на связь между токсичностью соединений и

их активностью. Корреляция между остальными парами признаков не обнаружена.

Вне зависимости от корреляционной картины при решении задач регрессии и классификации относительно определённой целевой величины другие целевые величины не будут использоваться в качестве признаков.

Был проведён корреляционный анализ прочих признаков с целевыми. Его результаты представлены на рисунках 9 – 11.

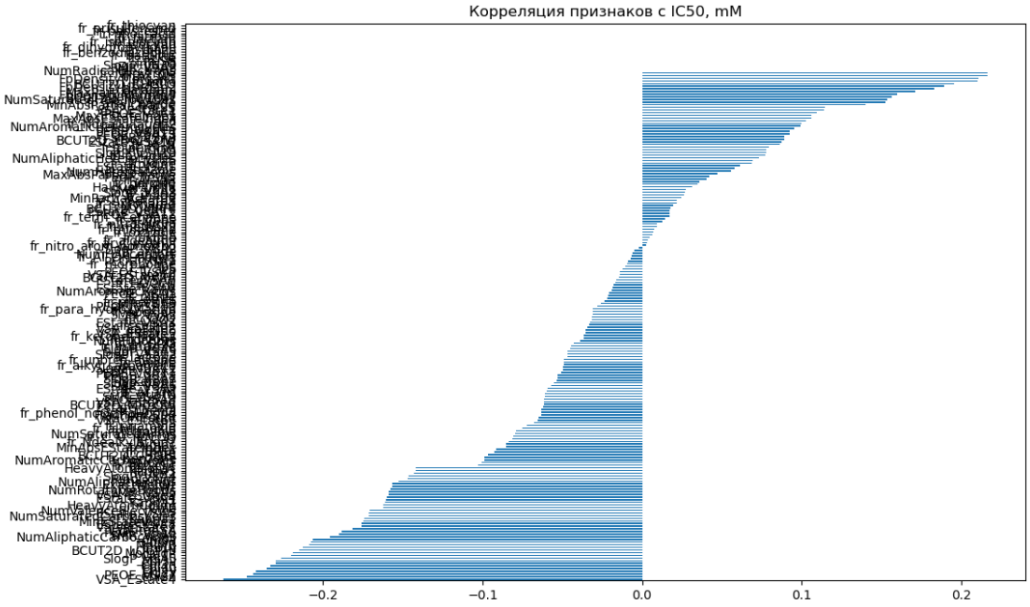

Рисунок 9 – корреляция признаков с IC_{50}

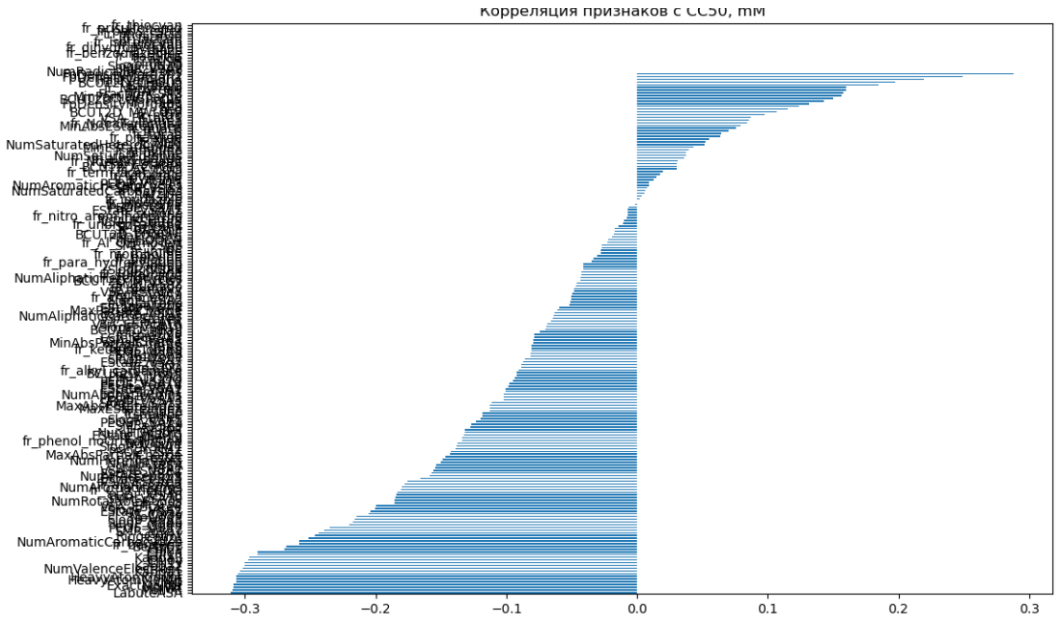

Рисунок 10 – корреляция всех признаков с CC_{50}

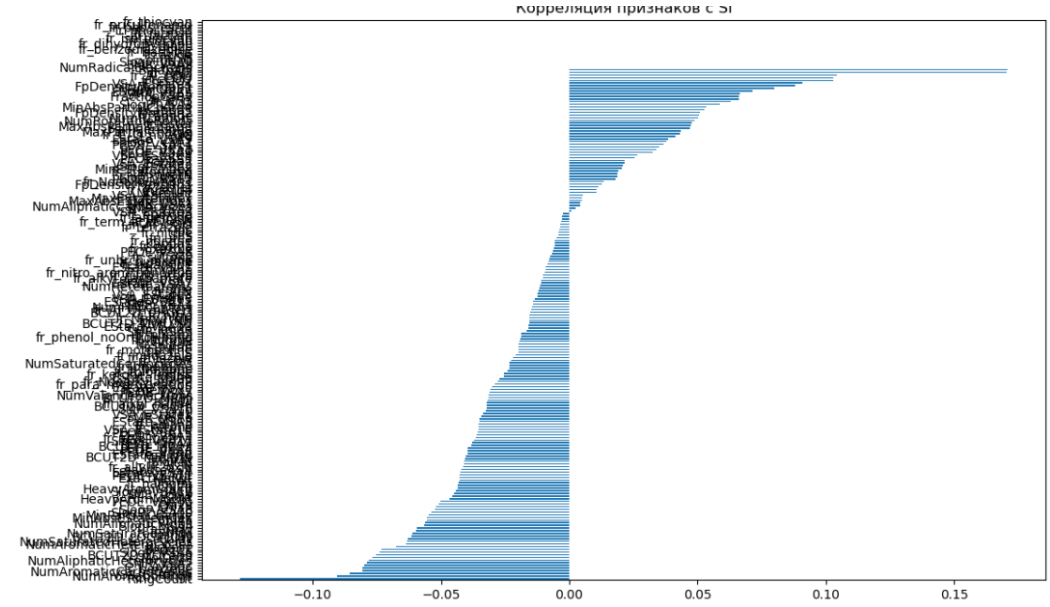

Рисунок 11 – корреляция всех признаков с SI

Как видно из барплотов, корреляция нецелевых признаков с целевыми либо прямая/обратная слабая, либо отсутствует.

Также был проведён корреляционный анализ для нецелевых признаков. Его результаты представлены в виде матрицы корреляции на рисунке 12.

Как видно из матрицы корреляции, большая часть признаков не имеет сильной корреляции друг с другом, однако наблюдаются пары признаков, имеющие сильную корреляцию и в некоторых случаях линейную зависимость.

Для устранения линейной зависимости и мультиколлинеарности необходимо удалить из каждой пар признаков с сильной корреляцией один из признаков.

Данный шаг будет выполнено на этапе отбора признаков.

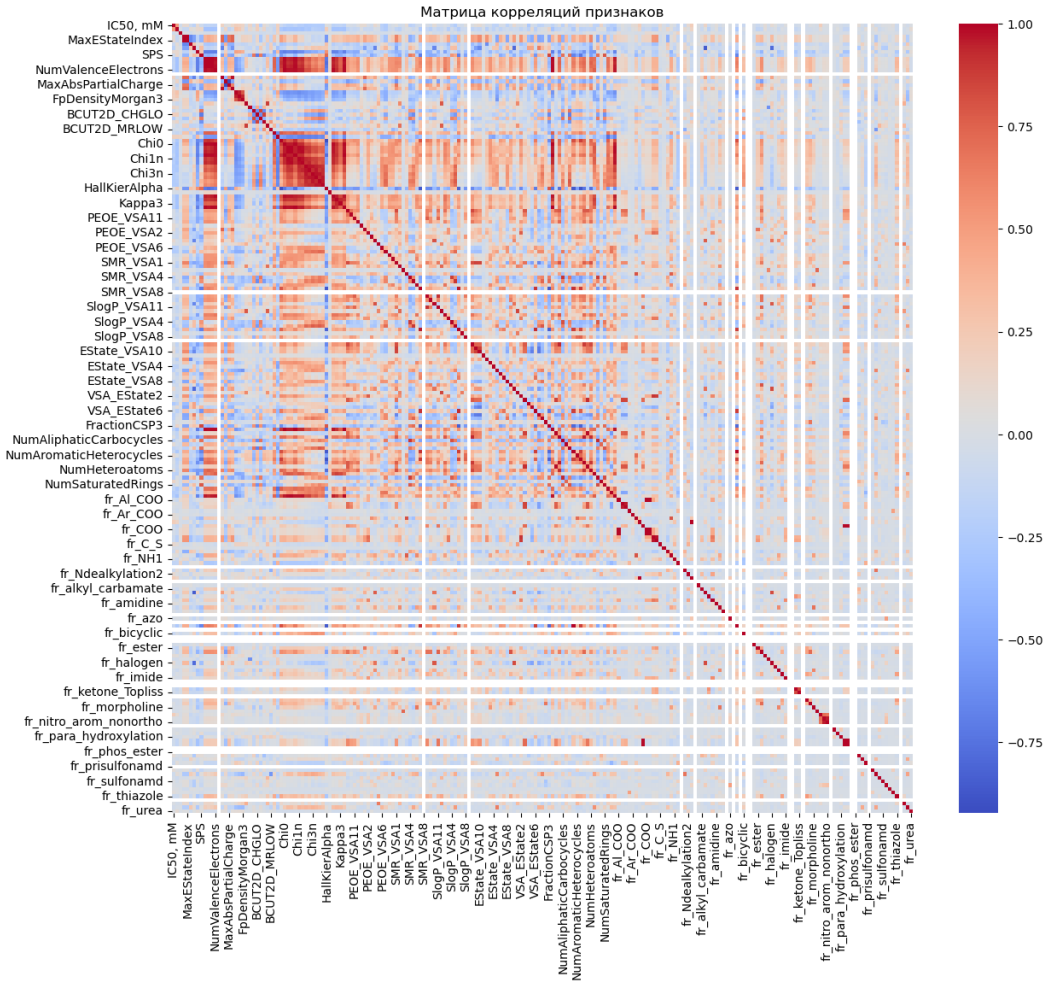

Рисунок 12 – матрица корреляций для нецелевых признаков

1.2 Предобработка данных

В рамках этапа предобработки данных проведена очистка датасета от признаков по результатам корреляционного анализа. Также проведены анализ выбросов и преобразования над признаками для улучшения качества работы моделей.

1.2.1 Анализ выбросов

Выбросы в среднем понижают устойчивость модели, что очевидно сказывается на результатах прогнозирования.

В рамках данного раздела были проведены работы по анализу выбросов.

Оценка производилась по межквартильному размаху, использовались 1-й и 3-й квартили.

В рамках данной методологии были проведены расчёты выбросов для основных целевых величин. Результаты представлены на рисунках 13 – 16:

```
Количество выбросов в каждом признаке:
IC50, mM: 140 выбросов
CC50, mM: 35 выбросов
SI: 119 выбросов
```

Рисунок 13 – численные значения выбросов по межквартильному размаху для целевых переменных

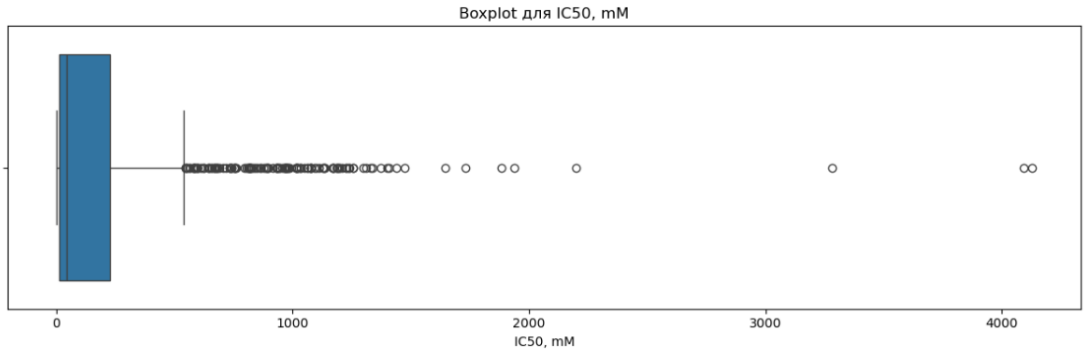

Рисунок 14 – boxplot для IC_{50}

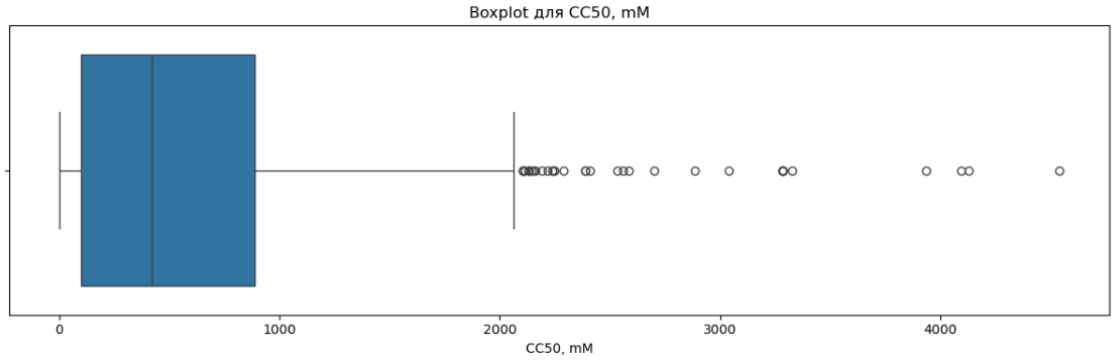

Рисунок 15 – boxplot для CC_{50}

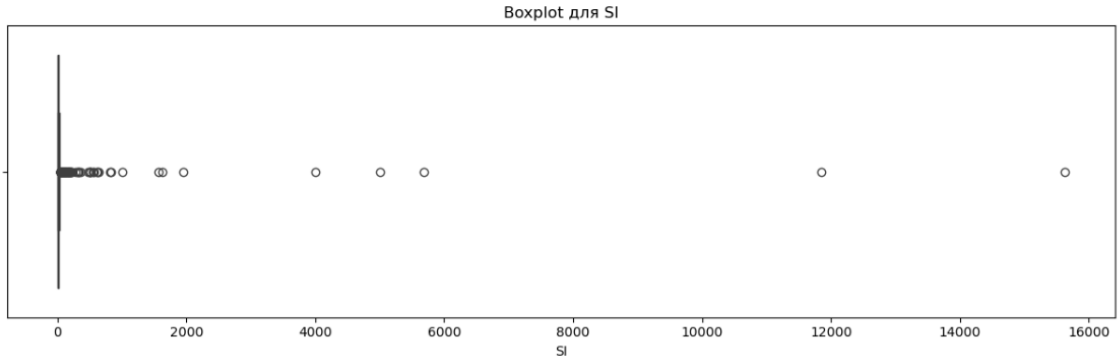

Рисунок 16 – boxplot для SI

Большинство из строк, в которых обнаруживаются выбросы, не пересекаются, ввиду чего было принято решение производить очистку выбросов непосредственно перед этапом обучения, ориентируясь на значение для каждой целевой величины.

Также было принято решение рассмотреть другую методологию оценки выбросов: по 3σ.

Аналогичный анализ был проведён для нецелевых признаков. Его результаты представлены на рисунке 17

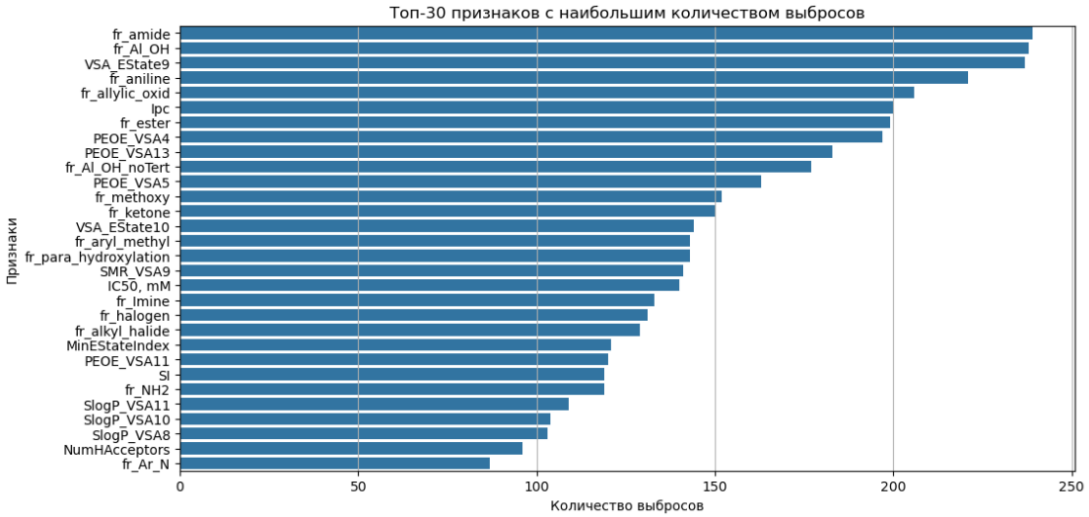

Рисунок 17 – 30 признаков с наибольшем количеством выбросов

Как видно из рисунка 17, не только целевые величины обладают большим количеством выбросов. Это может значить, что распределения признаков также имеют выраженную асимметрию и подлежат обработке (например, логарифмированию). Оставим данный тезис для предложений по улучшению результатов.

1.2.2 Отбор признаков

Отбор признаков производился на основе двух критериев: критерий нулевой дисперсии и результаты корреляционного анализа.

В рамках работы с первым критерием были определены и отброшены признаки, имеющие нулевую дисперсию. Их названия приведены на рисунке 18.

```
low_variance = df.var()[df.var() == 0].index
low_variance
```

```
Index(['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'],
      dtype='object')
```

Рисунок 18 – названия признаков с нулевой дисперсией

После удалению данных признаков в датасете осталось 195 полей, включая поля ключевых величин.

В рамках работы с критерием корреляционного анализа были отобраны пары с высокой корреляцией, характер связи между которыми тяготеет к линейной зависимости (коэффициент Пирсона > 0.9).

Данные пары признаков представлены ниже в формате Признак 1 и Признак 2: коэффициент Пирсона

1. MaxAbsEStateIndex и MaxEStateIndex: 1.00
2. MolWt и HeavyAtomMolWt: 1.00
3. MolWt и ExactMolWt: 1.00
4. MolWt и NumValenceElectrons: 0.98
5. MolWt и BertzCT: 0.90
6. MolWt и Chi0: 0.99
7. MolWt и Chi0n: 0.94
8. MolWt и Chi0v: 0.95
9. MolWt и Chi1: 0.99
10. MolWt и Chi1n: 0.90
11. MolWt и Chi1v: 0.93
12. MolWt и Kappa1: 0.96
13. MolWt и Kappa2: 0.91
14. MolWt и LabuteASA: 0.99
15. MolWt и HeavyAtomCount: 0.99
16. MolWt и MolMR: 0.96
17. HeavyAtomMolWt и ExactMolWt: 1.00
18. HeavyAtomMolWt и NumValenceElectrons: 0.97
19. HeavyAtomMolWt и BertzCT: 0.92
20. HeavyAtomMolWt и Chi0: 0.98
21. HeavyAtomMolWt и Chi0n: 0.91
22. HeavyAtomMolWt и Chi0v: 0.93

23.HeavyAtomMolWt и Chi1: 0.98

24.HeavyAtomMolWt и Kappa1: 0.94

25.HeavyAtomMolWt и LabuteASA: 0.98

26.HeavyAtomMolWt и HeavyAtomCount: 0.98

27.HeavyAtomMolWt и MolMR: 0.94

28.ExactMolWt и NumValenceElectrons: 0.98

29.ExactMolWt и BertzCT: 0.90

30.ExactMolWt и Chi0: 0.99

31.ExactMolWt и Chi0n: 0.94

32.ExactMolWt и Chi0v: 0.95

33.ExactMolWt и Chi1: 0.99

34.ExactMolWt и Chi1n: 0.91

35.ExactMolWt и Chi1v: 0.93

36.ExactMolWt и Kappa1: 0.96

37.ExactMolWt и Kappa2: 0.91

38.ExactMolWt и LabuteASA: 0.99

39.ExactMolWt и HeavyAtomCount: 0.99

40.ExactMolWt и MolMR: 0.96

41.NumValenceElectrons и Chi0: 0.99

42.NumValenceElectrons и Chi0n: 0.98

43.NumValenceElectrons и Chi0v: 0.97

44.NumValenceElectrons и Chi1: 0.98

45.NumValenceElectrons и Chi1n: 0.95

46.NumValenceElectrons и Chi1v: 0.95

47.NumValenceElectrons и Kappa1: 0.99

48.NumValenceElectrons и Kappa2: 0.93

49.NumValenceElectrons и LabuteASA: 0.99

50.NumValenceElectrons и HeavyAtomCount: 0.99

51.NumValenceElectrons и MolMR: 0.97

52.MaxPartialCharge и MinAbsPartialCharge: 0.97

53. MinPartialCharge и MaxAbsPartialCharge: -0.92
54. FpDensityMorgan1 и FpDensityMorgan2: 0.95
55. FpDensityMorgan2 и FpDensityMorgan3: 0.94
56. BertzCT и Chi1: 0.92
57. BertzCT и HallKierAlpha: -0.90
58. BertzCT и HeavyAtomCount: 0.91
59. Chi0 и Chi0n: 0.96
60. Chi0 и Chi0v: 0.96
61. Chi0 и Chi1: 0.99
62. Chi0 и Chi1n: 0.93
63. Chi0 и Chi1v: 0.93
64. Chi0 и Kappa1: 0.98
65. Chi0 и Kappa2: 0.92
66. Chi0 и LabuteASA: 0.99
67. Chi0 и HeavyAtomCount: 1.00
68. Chi0 и MolMR: 0.96
69. Chi0n и Chi0v: 0.99
70. Chi0n и Chi1: 0.95
71. Chi0n и Chi1n: 0.99
72. Chi0n и Chi1v: 0.98
73. Chi0n и Chi2n: 0.90
74. Chi0n и Kappa1: 0.97
75. Chi0n и LabuteASA: 0.97
76. Chi0n и HeavyAtomCount: 0.96
77. Chi0n и MolMR: 0.98
78. Chi0v и Chi1: 0.95
79. Chi0v и Chi1n: 0.98
80. Chi0v и Chi1v: 0.99
81. Chi0v и Chi2v: 0.91
82. Chi0v и Kappa1: 0.96

83.Chi0v и LabuteASA: 0.98

84.Chi0v и HeavyAtomCount: 0.96

85.Chi0v и MolMR: 0.99

86.Chi1 и Chi1n: 0.92

87.Chi1 и Chi1v: 0.93

88.Chi1 и Kappa1: 0.95

89.Chi1 и Kappa2: 0.91

90.Chi1 и LabuteASA: 0.99

91.Chi1 и HeavyAtomCount: 1.00

92.Chi1 и MolMR: 0.97

93.Chi1n и Chi1v: 0.98

94.Chi1n и Chi2n: 0.94

95.Chi1n и Chi2v: 0.92

96.Chi1n и Kappa1: 0.94

97.Chi1n и LabuteASA: 0.95

98.Chi1n и HeavyAtomCount: 0.93

99.Chi1n и MolMR: 0.97

100. Chi1v и Chi2n: 0.91

101. Chi1v и Chi2v: 0.94

102. Chi1v и Kappa1: 0.93

103. Chi1v и LabuteASA: 0.96

104. Chi1v и HeavyAtomCount: 0.93

105. Chi1v и MolMR: 0.98

106. Chi2n и Chi2v: 0.97

107. Chi2n и Chi3n: 0.97

108. Chi2n и Chi3v: 0.95

109. Chi2n и Chi4n: 0.93

110. Chi2n и Chi4v: 0.91

111. Chi2v и Chi3n: 0.93

112. Chi2v и Chi3v: 0.97

113. Chi2v и Chi4v: 0.93
114. Chi3n и Chi3v: 0.97
115. Chi3n и Chi4n: 0.96
116. Chi3n и Chi4v: 0.93
117. Chi3v и Chi4n: 0.94
118. Chi3v и Chi4v: 0.97
119. Chi4n и Chi4v: 0.97
120. Kappa1 и Kappa2: 0.96
121. Kappa1 и LabuteASA: 0.97
122. Kappa1 и HeavyAtomCount: 0.96
123. Kappa1 и MolMR: 0.95
124. Kappa2 и Kappa3: 0.94
125. Kappa2 и LabuteASA: 0.91
126. Kappa2 и HeavyAtomCount: 0.91
127. Kappa2 и MolMR: 0.91
128. LabuteASA и HeavyAtomCount: 0.99
129. LabuteASA и MolMR: 0.99
130. SMR_VSA7 и SlogP_VSA6: 0.96
131. SMR_VSA7 и VSA_EState6: 0.90
132. SMR_VSA7 и NumAromaticCarbocycles: 0.91
133. SMR_VSA7 и fr_benzene: 0.91
134. SMR_VSA9 и SlogP_VSA11: 0.91
135. SlogP_VSA6 и VSA_EState6: 0.92
136. TPSA и NOCount: 0.94
137. VSA_EState2 и fr_C_O: 0.90
138. VSA_EState3 и NumHDonors: 0.92
139. HeavyAtomCount и MolMR: 0.97
140. NHOHCount и NumHDonors: 0.98
141. NOCount и NumHAcceptors: 0.96
142. NOCount и NumHeteroatoms: 0.92

143. NumAliphaticCarbocycles и NumSaturatedCarbocycles: 0.93
144. NumAromaticCarbocycles и fr_benzene: 1.00
145. fr_Al_COO и fr_COO: 0.99
146. fr_Al_COO и fr_COO2: 0.99
147. fr_Al_OH и fr_Al_OH_noTert: 0.96
148. fr_Ar_NH и fr_Nhpyrrole: 1.00
149. fr_Ar_OH и fr_phenol: 0.99
150. fr_Ar_OH и fr_phenol_noOrthoHbond: 0.99
151. fr_COO и fr_COO2: 1.00
152. fr_C_O и fr_C_O_noCOO: 0.98
153. fr_nitro_arom и fr_nitro_arom_nonortho: 0.96
154. fr_phenol и fr_phenol_noOrthoHbond: 1.00

Была сформирована хэш-таблица, в которой было подсчитано, сколько раз каждый признак встречается с другим в рамках приведённых 154 пар:

```
{'MolWt': 15,
 'ExactMolWt': 15,
 'Chi1': 15,
 'HeavyAtomCount': 15,
 'NumValenceElectrons': 14,
 'Chi0': 14,
 'Chi0n': 14,
 'Chi0v': 14,
 'Chi1n': 14,
 'Chi1v': 14,
 'Kappa1': 14,
 'LabuteASA': 14,
 'MolMR': 14,
 'HeavyAtomMolWt': 12,
 'Kappa2': 10,
 'Chi2n': 8,
 'Chi2v': 7,
 'BertzCT': 6,
 'Chi3n': 5,
 'Chi3v': 5,
 'Chi4v': 5,
 'Chi4n': 4,
 'SMR_VSA7': 4,
 'NOCount': 3,
 'FpDensityMorgan2': 2,
 'SlogP_VSA6': 2,
 'VSA_EState6': 2,
 'NumAromaticCarbocycles': 2,
 'fr_benzene': 2,
 'fr_C_O': 2,
```

```
'NumHDonors': 2,
'fr_Al_COO': 2,
'fr_COO': 2,
'fr_COO2': 2,
'fr_Ar_OH': 2,
'fr_phenol': 2,
'fr_phenol_noOrthoHbond': 2,
'MaxAbsEStateIndex': 1,
'MaxEStateIndex': 1,
'MaxPartialCharge': 1,
'MinAbsPartialCharge': 1,
'MinPartialCharge': 1,
'MaxAbsPartialCharge': 1,
'FpDensityMorgan1': 1,
'FpDensityMorgan3': 1,
'HallKierAlpha': 1,
'Kappa3': 1,
'SMR_VSA9': 1,
'SlogP_VSA11': 1,
'TPSA': 1,
'VSA_EState2': 1,
'VSA_EState3': 1,
'NHOHCount': 1,
'NumHAcceptors': 1,
'NumHeteroatoms': 1,
'NumAliphaticCarbocycles': 1,
'NumSaturatedCarbocycles': 1,
'fr_Al_OH': 1,
'fr_Al_OH_noTert': 1,
'fr_Ar_NH': 1,
'fr_Nhpyrrole': 1,
'fr_C_O_noCOO': 1,
'fr_nitro_arom': 1,
'fr_nitro_arom_nonortho': 1}
```

Было принято решение удалять правый признак из пары. В результате в датасете осталось 148 полей, включая поля целевых признаков.

Итого из 211 признаков в итоговую выборку попали 145, не включая целевые величины.

1.2.3 Нормализация данных

Многие модели чувствительны к тому, чтобы данные были нормализованы. Для того, чтобы не ограничивать себя в выборе моделей проведём нормализацию признаков при помощи StandardScaler.

Результаты его работы отражены на рисунке 19

```
features = df.columns[3:]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(df[features])
df_scaled = pd.DataFrame(scaled_features, columns=features)
df_scaled[targets] = df[targets].values
```

```
df_scaled.head()
```

	MaxAbsEStateIndex	MinAbsEStateIndex	MinEStateIndex	qed	SPS	MolWt	MaxPartialCharge	MinPartialCharge	FpDe
0	-1.763647	1.227715	0.858123	-0.755891	1.073323	0.258081	-1.556297	1.522921	
1	-2.108224	2.095440	0.951585	-0.544168	1.254983	0.289724	-1.759553	1.260180	
2	-2.514136	2.150839	0.957552	-1.490120	1.014426	0.745889	-1.118121	1.099403	
3	-1.762654	1.247702	0.860276	-0.941357	0.988562	0.368166	-1.556297	1.522921	
4	-1.746485	0.536885	0.783714	-0.701088	0.563542	0.902106	-1.367945	2.002451	

5 rows × 148 columns

Рисунок 19 – нормализация данных

Выводы

Таким образом, в рамках EDA были проведены следующие операции и получены следующие результаты:

1. Проведены первичный анализ и знакомство с данными.
2. Удалены дубликаты.
3. Удалены строки с пустыми значениями.
4. Определён характер распределения целевых величин: левоассиметричное распределение с большим количеством выбросов. Сделано предположение о положительном влиянии логарифмирования целевых переменных IC_{50} и CC_{50} на результаты работы моделей МО.
5. Проведён корреляционный анализ целевых и нецелевых признаков. Определены пары признаков с сильной корреляцией.
6. Проведены анализ выбросов, сформированы критерии для удаления выбросов для каждой из целевых переменных.
7. Проведена нормализация признаков.
8. Произведено сохранение итогового набор данных в формате .parquet.

2 Решение задачи классификации

В рамках задачи классификации необходимо обучить бинарный классификатор, способный с достаточной точностью прогнозировать принадлежность одной из целевых величин к одному из заданных классов.

Задачи классификации в рамках данной курсовой работы в целом похожи и намеренно решались при помощи идентичных инструментов с целью оценки их различий.

Для решения задач классификации был разработан следующий план:

1. Выделение целевых классов согласно условию;
2. Проверка на дисбаланс классов;
3. Удаление выбросов;
4. Обучение и базовая проверка бейзлайна;
5. Оптимизации избранных моделей из бейзлайна;
6. Анализ результатов работы моделей;
7. Сравнение моделей и выбор лучшей.

Каждая из реализаций задач соответствует данному плану.

2.1 Классификация IC_{50}

В рамках данной задачи выполнялась работа по построению классификатора для определения принадлежности метрики IC_{50} к множеству значений, каждое из которых больше медианного значения по всей выборке.

2.1.1 Предобработка данных

После загрузки данных была сформирована целевая величина (рисунок 20):

```
df["IC50_gt_median"] = (df[target] > df[target].median()).astype(int)
```

Рисунок 20 – получение целевой величины

```
class_ratio = np.mean(y_train_ic50)
print(f"Баланс классов: {class_ratio:.2f} / {1-class_ratio:.2f}")
```

```
Баланс классов: 0.49 / 0.51
```

Рисунок 21 – отношение меток классов

Классы распределены практически поровну, что позволяет нам не задумываться о методах работы с дисбалансом классов.

После этого проведено удаление выбросов. В данном случае использовано правило 3σ, поскольку межквартильный размах даёт слишком много выбросов и совершенно точно отсекает важные значения. Результаты приведены на рисунке 22.

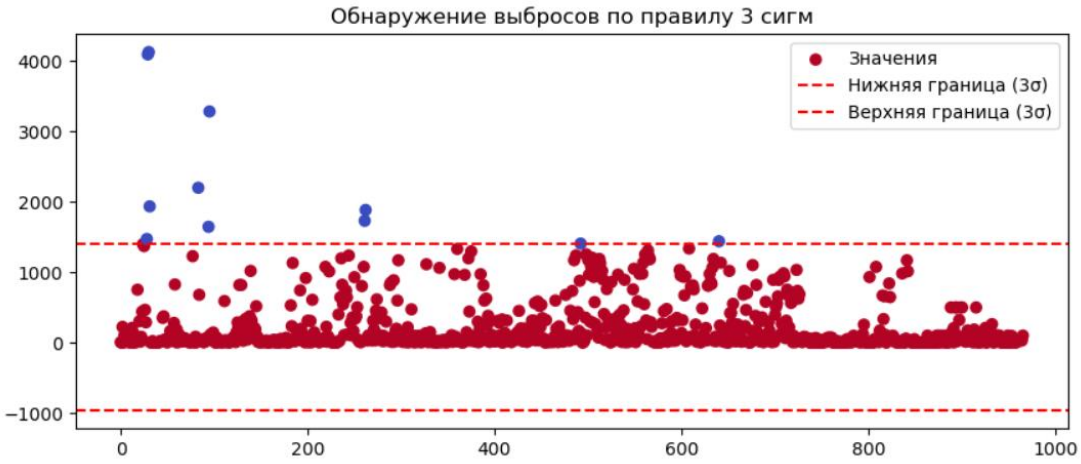

Рисунок 22 – отсечённые выбросы

После отсечения в датасете осталось 966 строк, что вполне достаточно для обучения без аугментации данных.

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета. Такой объём тестовой выборки был выбран в виду того, что данных не слишком много и уменьшение объёма обучающей выборки может отрицательно сказаться на обучении модели.

2.1.2 Бейзлайн

В качестве бейзлайна были выбраны следующие модели:

1. Логистическая регрессия – линейная модель для бинарной/многоклассовой классификации. Логистическая регрессия быстро обучается, однако требует масштабирования данных и плохо себя показывает при работе с нелинейными зависимостями.
2. Random Forest Classifier – Ансамбль деревьев решений с бэггингом. Устойчива к переобучению, хорошо показывает себя при работе с нелинейными зависимостями.
3. XGBoost Classifier – Градиентный бустинг над деревьями. Модель может давать высокую точность, а также обрабатывать разреженные данные. В целях снижения переобучения обладает встроенной регурялизацией. Требует тщательной настройки гиперпараметров.

В качестве метрик были выбраны следующие:

1. ROC-AUC – Площадь под ROC-кривой (AUC-ROC). Измеряет способность модели различать классы при разных порогах. Чем выше AUC, тем лучше модель различает классы.
2. F1 – Гармоническое среднее precision и recall.
3. Precision – Доля правильных положительных предсказаний среди всех предсказанных положительных.
4. Recall – Доля правильных положительных предсказаний среди всех реальных положительных примеров.

Было принято, что в рамках данной задачи важным является не пропускать активные/токсичные вещества. Следовательно, акцент при выборе модели будет делаться на ROC-AUC и Recall.

Для обучение всех моделей была использована кросс-валидация типа Stratified K-Fold.

Результаты бейзлайна приведены на рисунке 23

	model	cv_mean_roc_auc	cv_std_roc_auc	ROC-AUC	F1	PRECISION	RECALL
0	logreg	0.754305	0.033643	0.690044	0.657895	0.632911	0.684932
1	rf	0.761141	0.036286	0.763554	0.685714	0.716418	0.657534
2	xgb	0.740188	0.030127	0.772815	0.690647	0.727273	0.657534

Рисунок 23 – метрики бейзлайна

Как видно, все модели показывают себя достойно.

2.1.3 Оптимизация моделей

В рамках оптимизации для трёх типов моделей, фигурирующих в бейзлайне, были подобраны гиперпараметры при помощи Optuna.

Для моделей были установлены следующие параметры:

- Количество итераций поиска (trial): 100
- Количество фолдов кросс-валидации: 10

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 24

```
Final Metrics on Test Set:
roc_auc: 0.6926
f1: 0.6250
precision: 0.6338
recall: 0.6164
accuracy: 0.6250
```

Рисунок 24 – результаты работы лучшей модели логистической регрессии на тестовой выборке

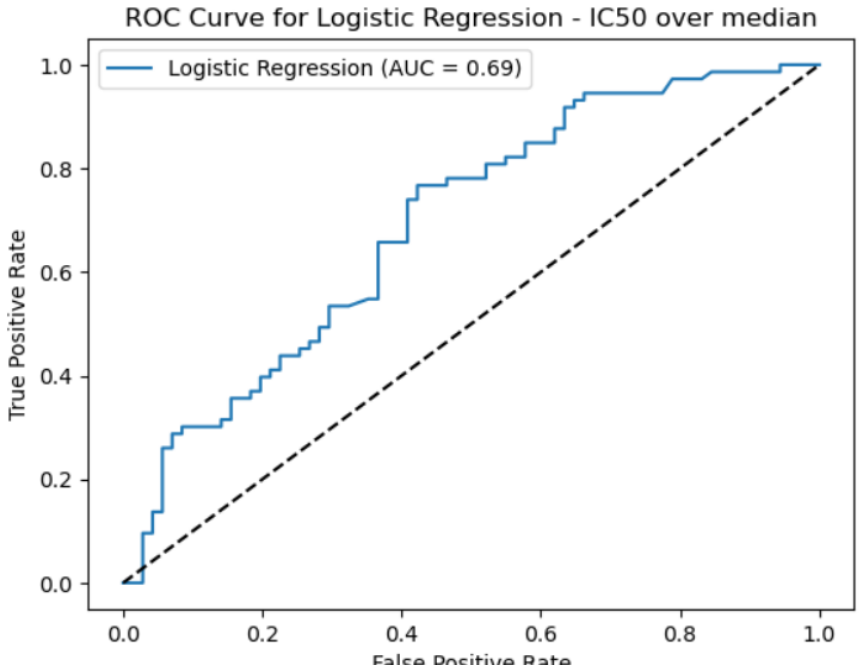

Рисунок 25 – ROC кривая для логистической регрессии

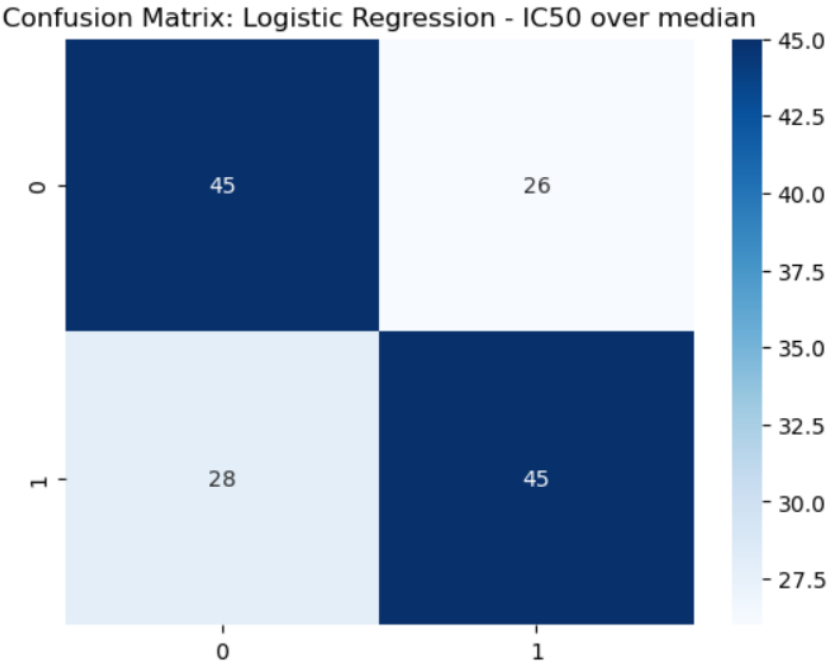

Рисунок 26 – матрица ошибок для логистической регрессии

```
Final Metrics on Test Set:
roc_auc: 0.7724
f1: 0.6525
precision: 0.6765
recall: 0.6301
accuracy: 0.6597
```

Рисунок 27 – результаты работы лучшей модели Random Forest на тестовой выборке

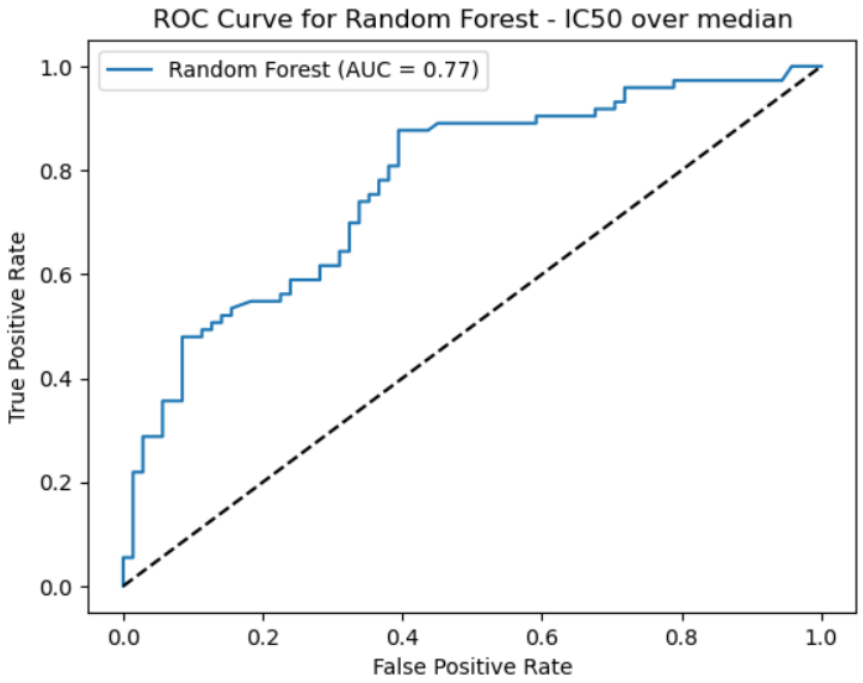

Рисунок 28 – ROC-кривая для Random Forest

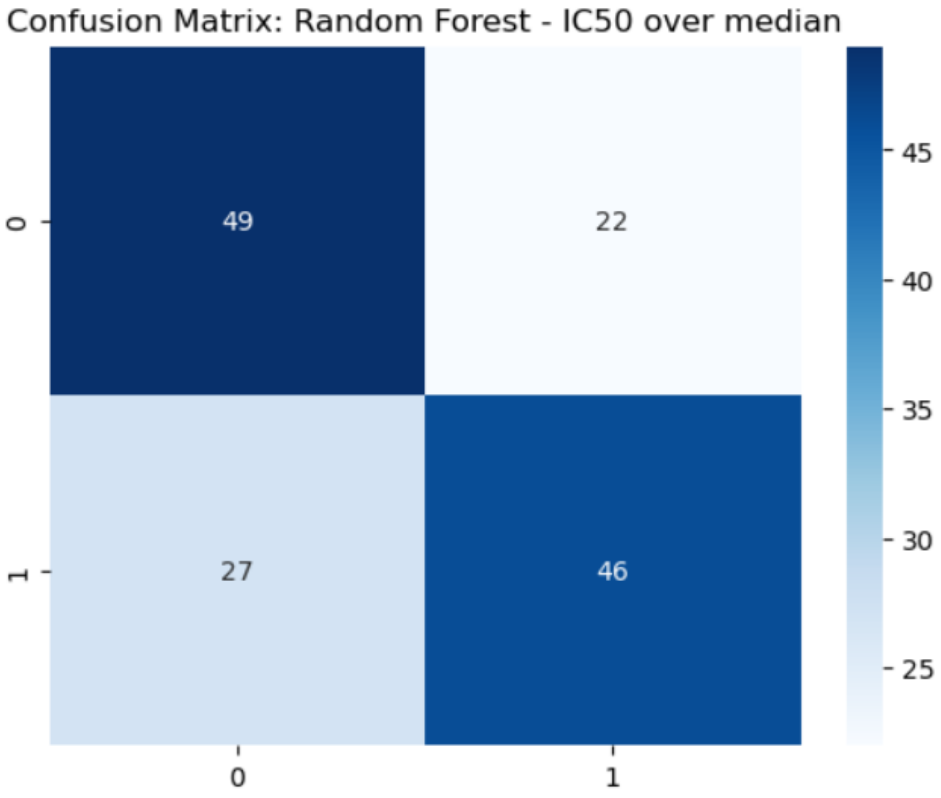

Рисунок 29 – матрица ошибок для Random Forest

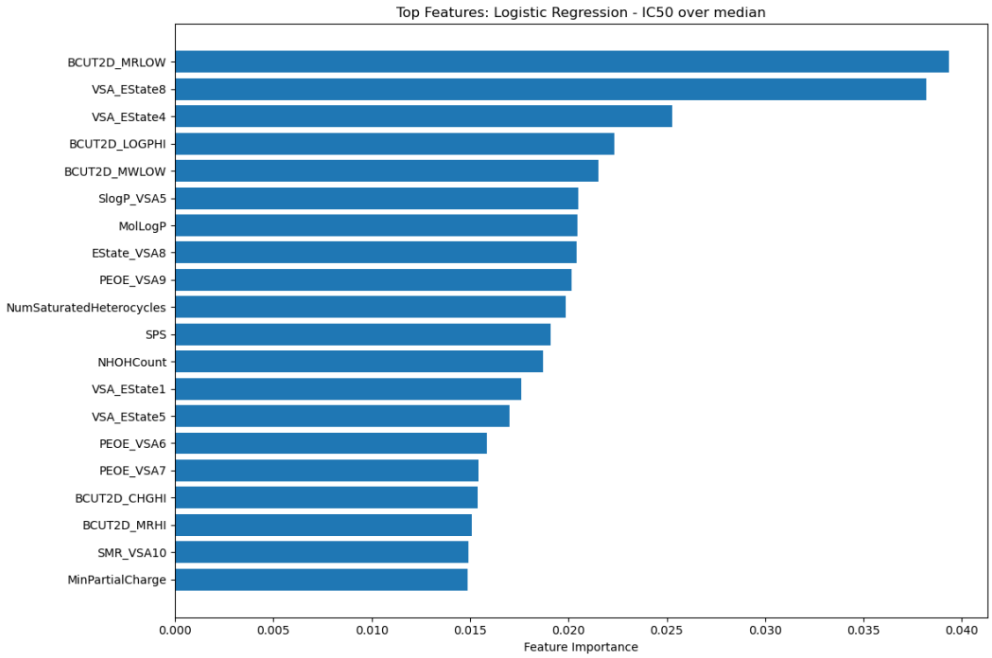

Рисунок 30 – важность признаков для Random Forest

```
Final Metrics on Test Set:
roc_auc: 0.7724
f1: 0.6525
precision: 0.6765
recall: 0.6301
accuracy: 0.6597
```

Рисунок 31 – результаты работы лучшей модели XGBoost на тестовой выборке

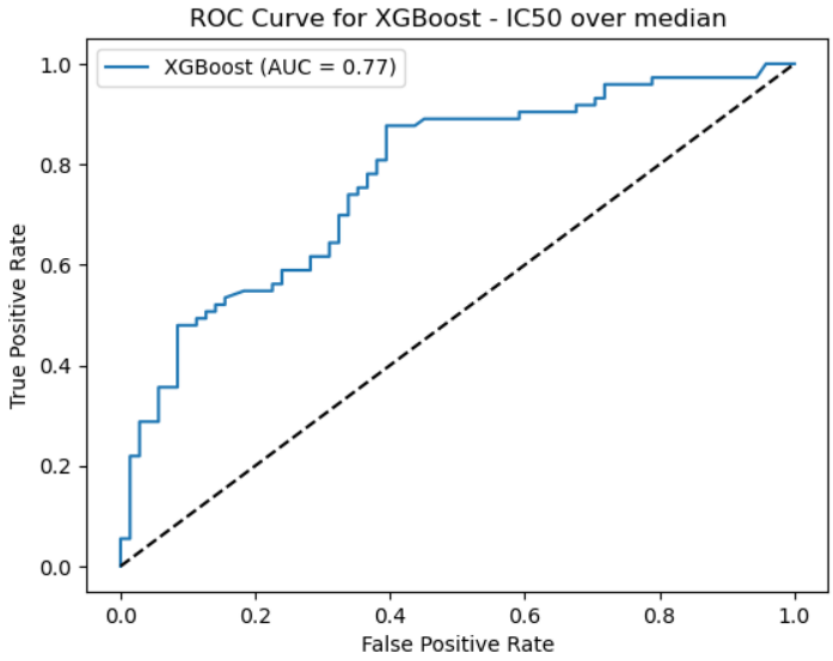

Рисунок 32 – ROC-кривая для XGBoost

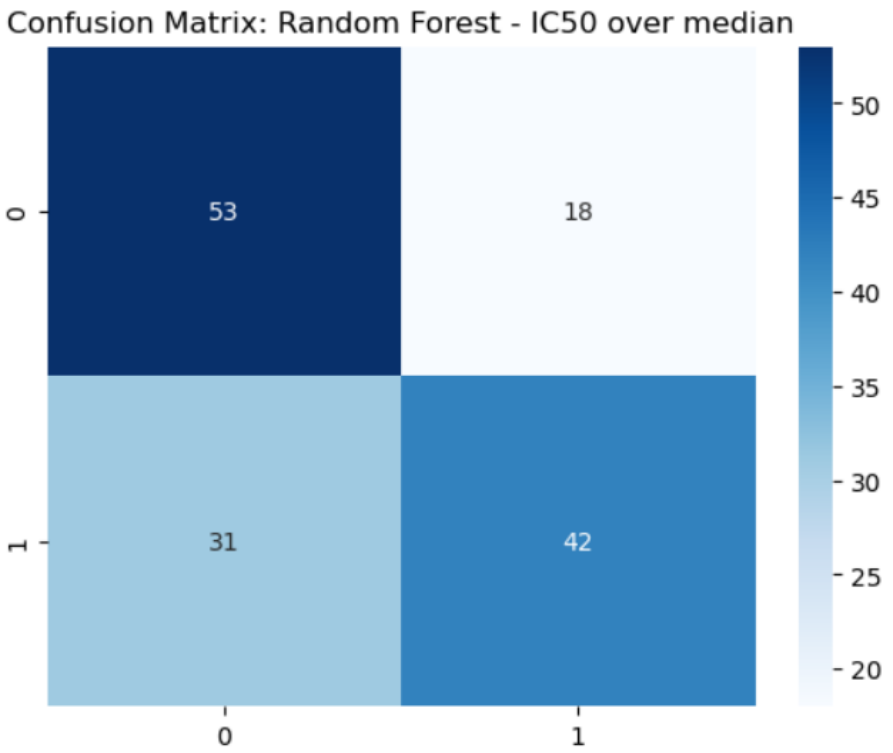

Рисунок 33 – матрица ошибок для XGBoost

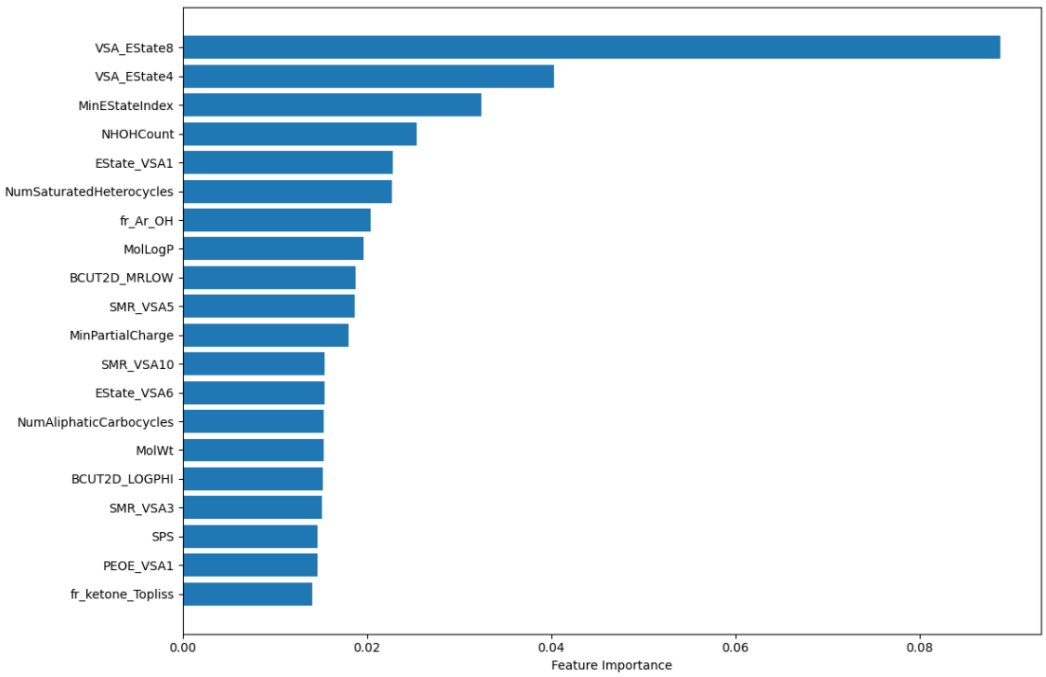

Рисунок 34 – важные признаки для XGBoost

2.1.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики ROC-AUC и Recall

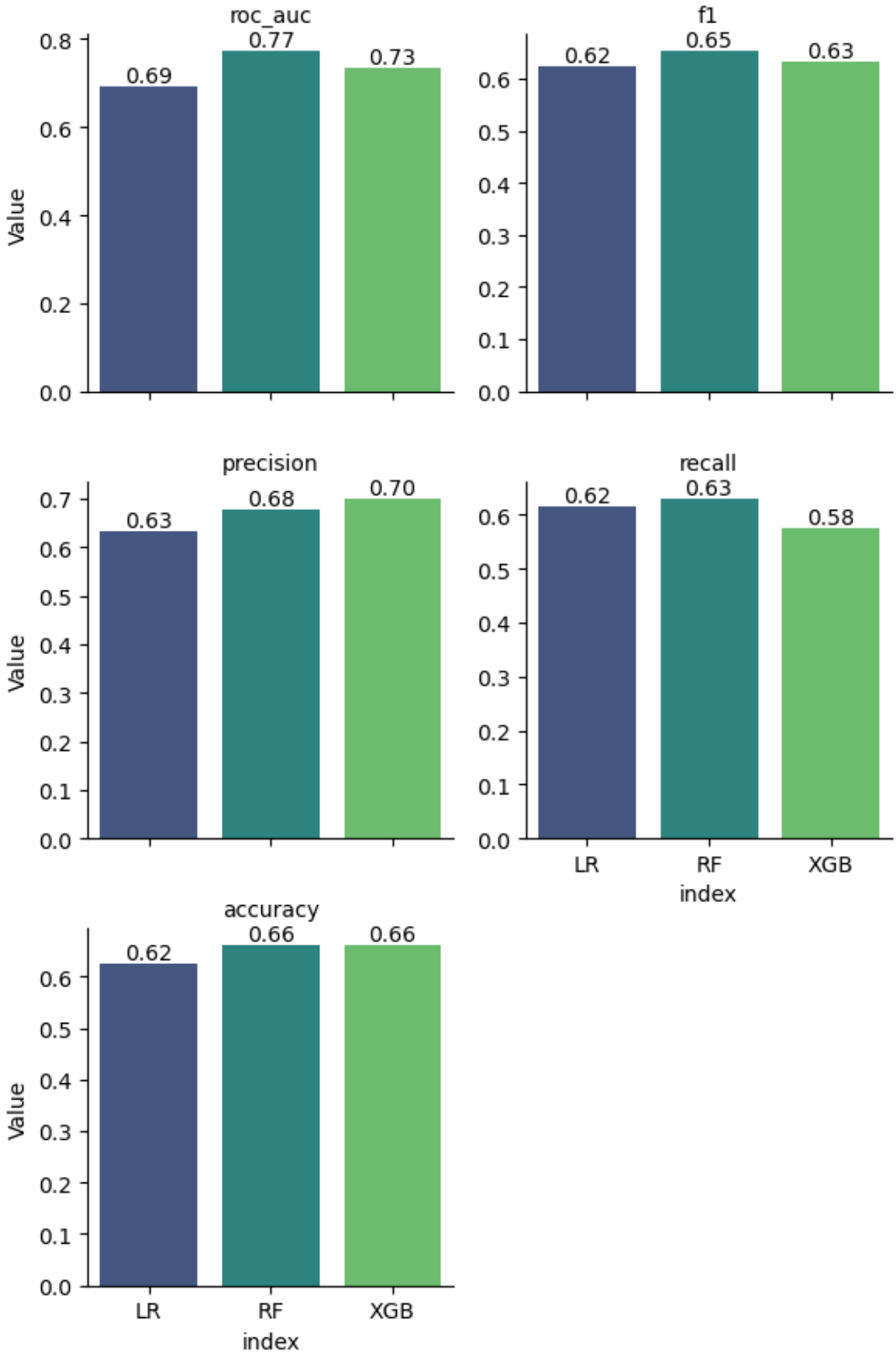

Рисунок 35 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 35, модели в целом близки по значениям. Явно выбивается логистическая регрессия, предположительно из-за наличия нелинейных зависимостей в признаках.

Random Forrest и XGBoost в целом сопоставимые модели в рамках данной задачи. Random Forest демонстрирует больший баланс между precision и recall, кроме того, среди всех испытанных моделей демонстрирует самый высокий Recall, что мы определились считать важным. Также Random Forest демонстрирует наиболее высокую ROC-AUC.

Следовательно, в рамках данной задачи Random Forest является предпочтительным вариантом.

Выводы

Задача классификации IC_{50} была решена.

Были проведены следующие этапы:

1. Предобработка данных, получение целевой переменной, очистка датасета от выбросов.
2. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей.
3. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.
4. Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками ROC-AUC и Recall.

2.2 Классификация CC_{50}

Задача классификации CC_{50} решалась аналогично задаче классификации IC_{50}, однако ещё с большим внимание к выбору модели относительно метрики Recall ввиду физического смысла целевой величины.

2.2.1 Предобработка данных

После загрузки данных была сформирована целевая величина (рисунок 36):

```
df["CC50_gt_median"] = (df[target] > df[target].median()).astype(int)
```

Рисунок 36 – получение целевой величины

```
class_ratio = np.mean(y_train_ic50)
print(f"Баланс классов: {class_ratio:.2f} / {1-class_ratio:.2f}")
```

```
Баланс классов: 0.50 / 0.50
```

Рисунок 37 – отношение меток классов

Классы распределены поровну.

После этого проведено удаление выбросов. В данном случае использовано правило 3σ. Результаты приведены на рисунке 38

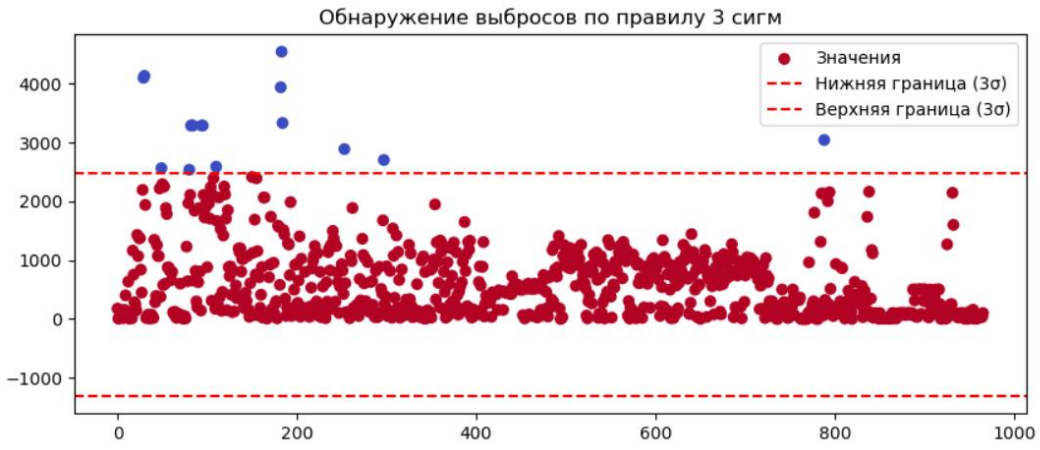

Рисунок 38 – отсечённые выбросы

После отсечения в датасете осталось 950 строк

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета.

2.2.2 Бейзлайн

В качестве бейзлайна были выбраны модели аналогичные моделям из задачи классификации IC_{50}

Для обучение всех моделей была использована кросс-валидация типа Stratified K-Fold.

Результаты бейзлайна приведены на рисунке 39

	model	cv_mean_roc_auc	cv_std_roc_auc	ROC-AUC	F1	PRECISION	RECALL
0	logreg	0.805219	0.028651	0.789245	0.666667	0.626866	0.711864
1	rf	0.829778	0.027205	0.819613	0.700855	0.706897	0.694915
2	xgb	0.811805	0.025622	0.793684	0.689076	0.683333	0.694915

Рисунок 39 – метрики бейзлайна

Как видно, все модели показывают себя достойно.

2.2.3 Оптимизация моделей

Оптимизация проводилась аналогично задаче IC_{50}.

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 40 - 51

```
Final Metrics on Test Set:
roc_auc: 0.7856
f1: 0.6565
precision: 0.5972
recall: 0.7288
accuracy: 0.6853
```

Рисунок 40 – результаты работы лучшей модели логистической регрессии на тестовой выборке

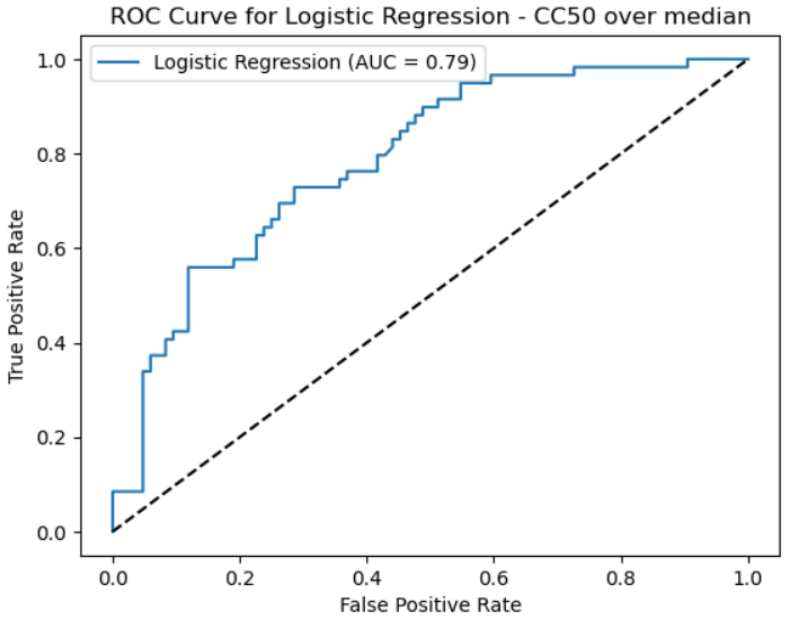

Рисунок 41 – ROC кривая для логистической регрессии

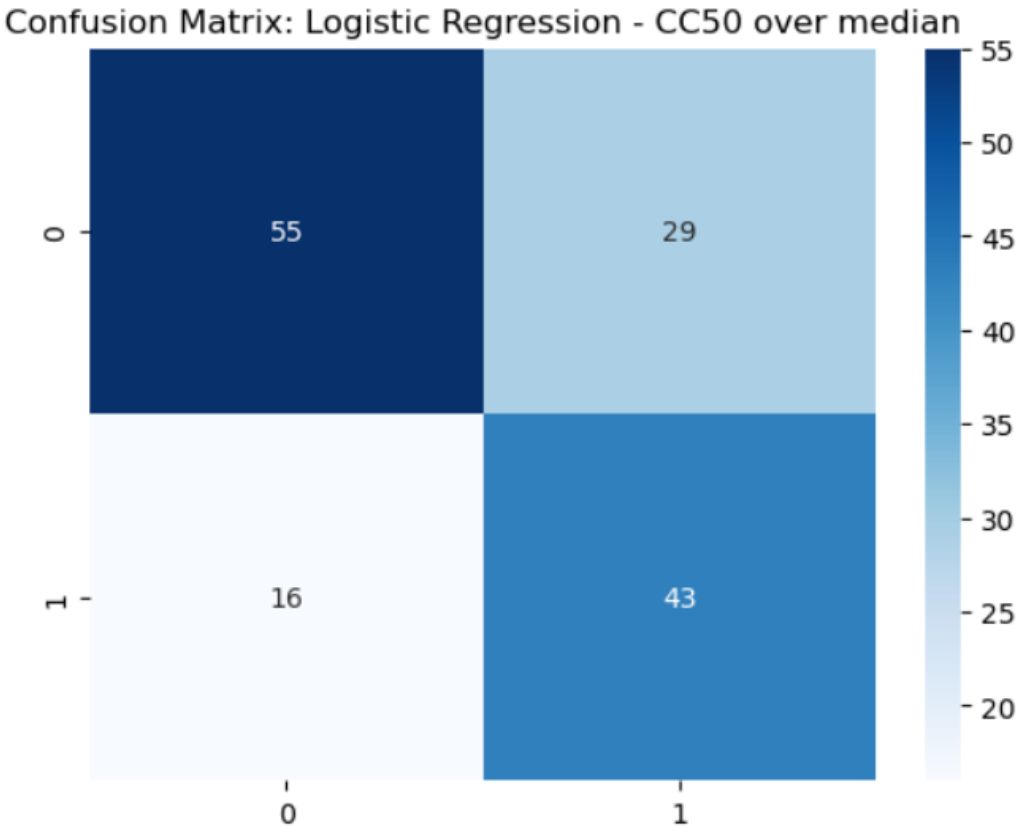

Рисунок 42 – матрица ошибок для логистической регрессии

```
Final Metrics on Test Set:
roc_auc: 0.7724
f1: 0.6525
precision: 0.6765
recall: 0.6301
accuracy: 0.6597
```

Рисунок 43 – результаты работы лучшей модели Random Forest на тестовой выборке

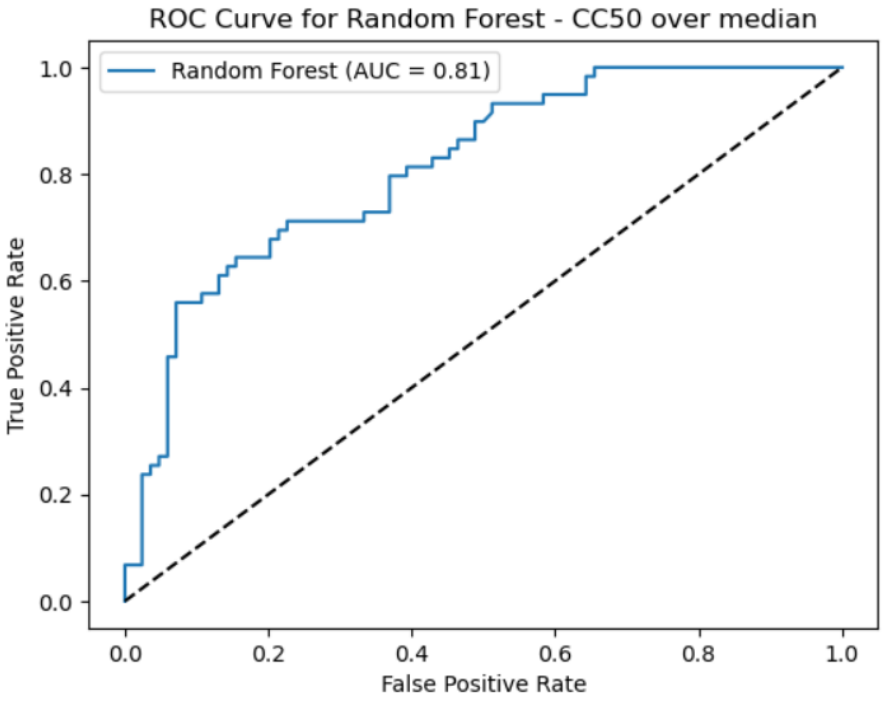

Рисунок 44 – ROC-кривая для Random Forest

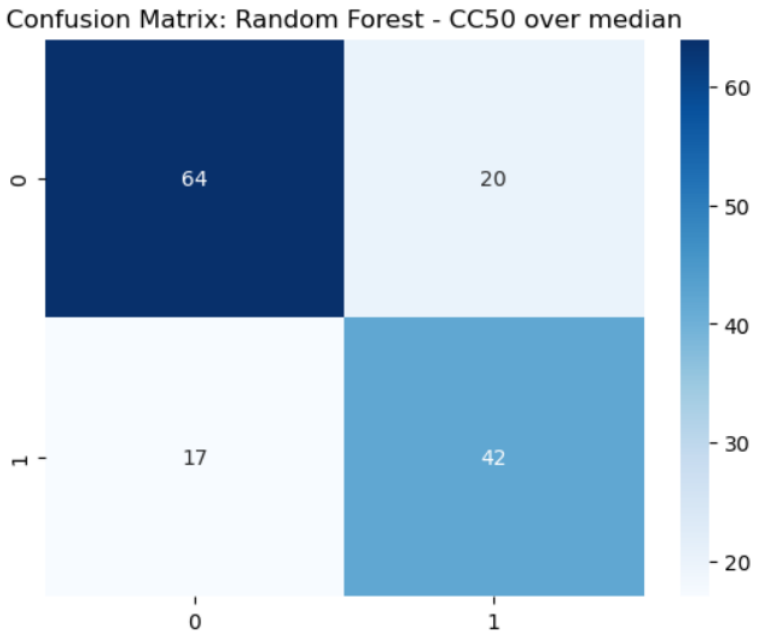

Рисунок 45 – матрица ошибок для Random Forest

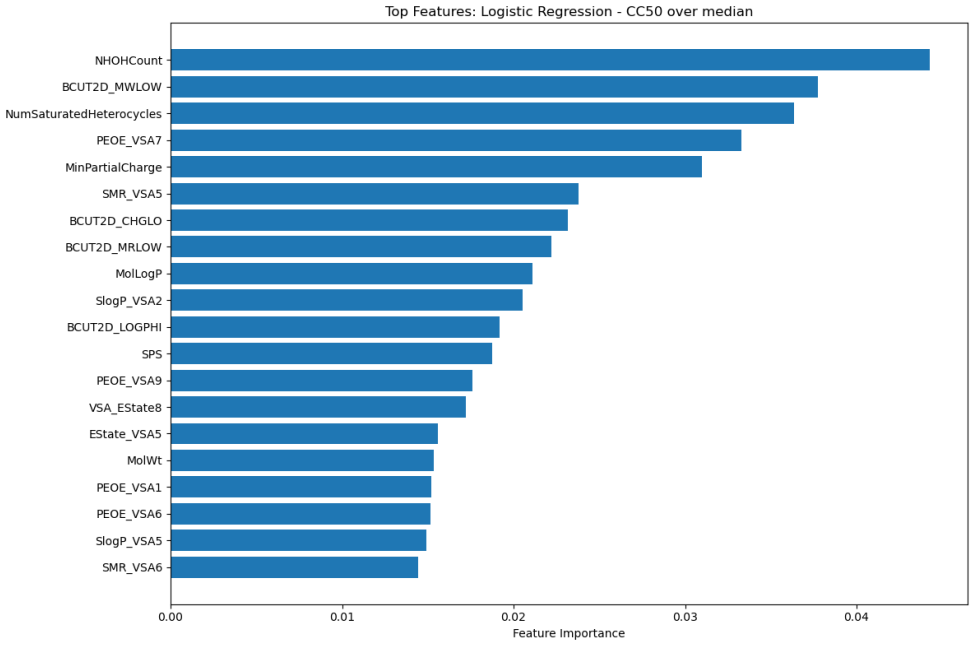

Рисунок 46 – важность признаков для Random Forest

```
Final Metrics on Test Set:
roc_auc: 0.8139
f1: 0.6942
precision: 0.6774
recall: 0.7119
accuracy: 0.7413
```

Рисунок 47 – результаты работы лучшей модели XGBoost на тестовой выборке

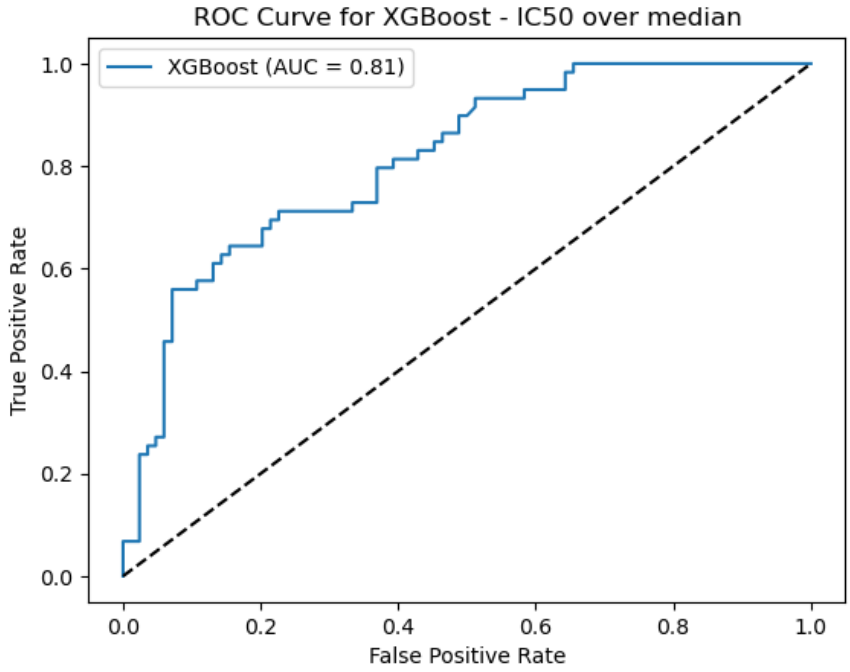

Рисунок 48 – ROC-кривая для XGBoost

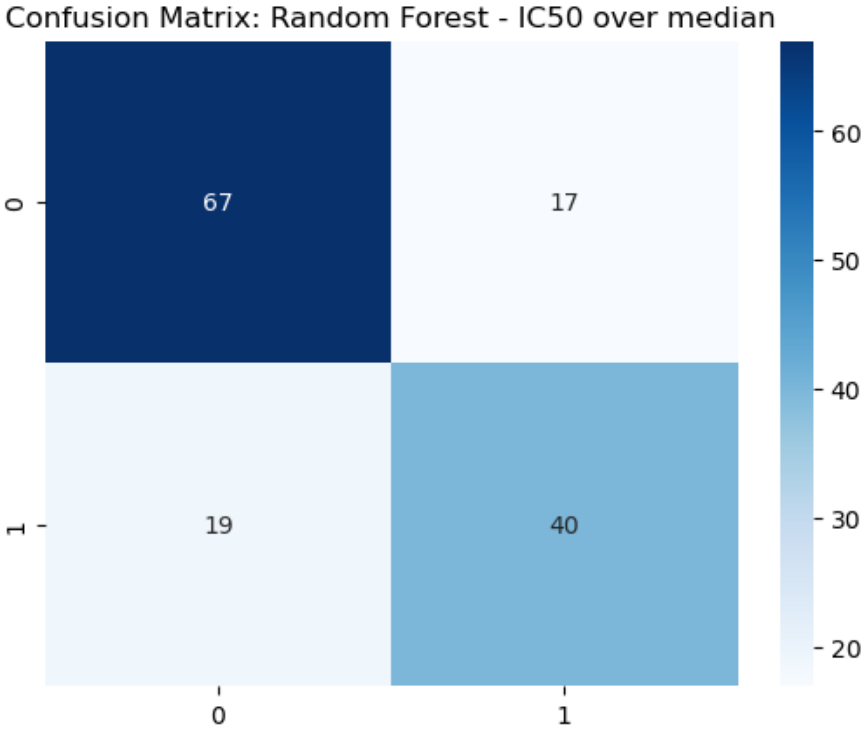

Рисунок 49 – матрица ошибок для XGBoost

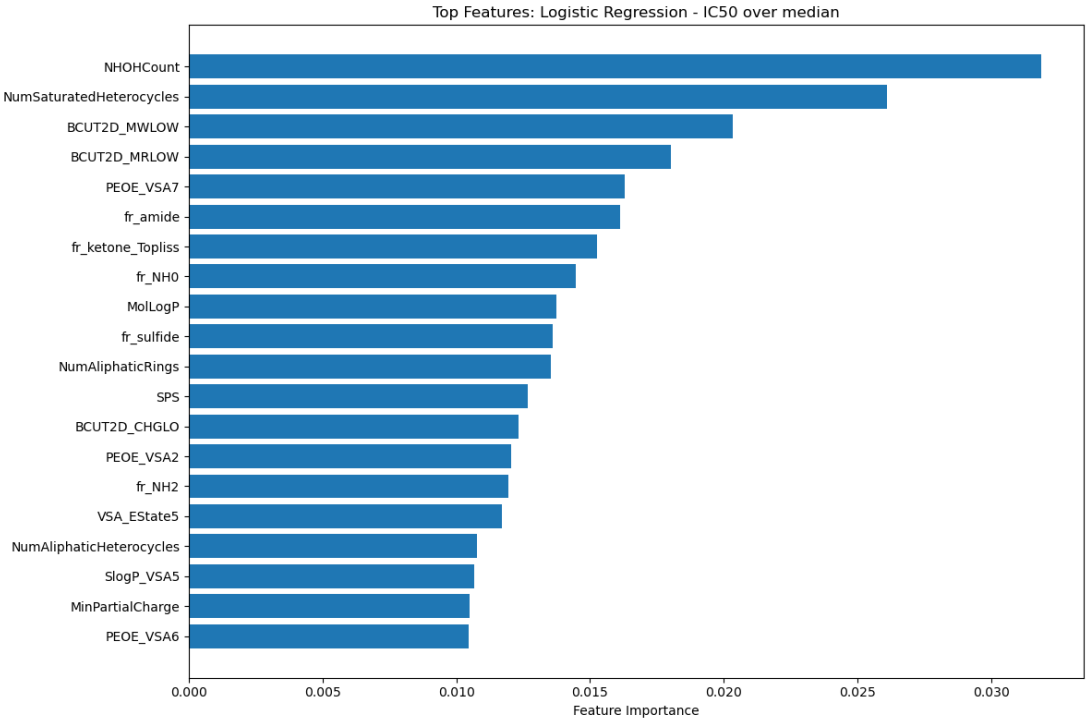

Рисунок 50 – важные признаки для XGBoost

2.2.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики ROC-AUC и Recall

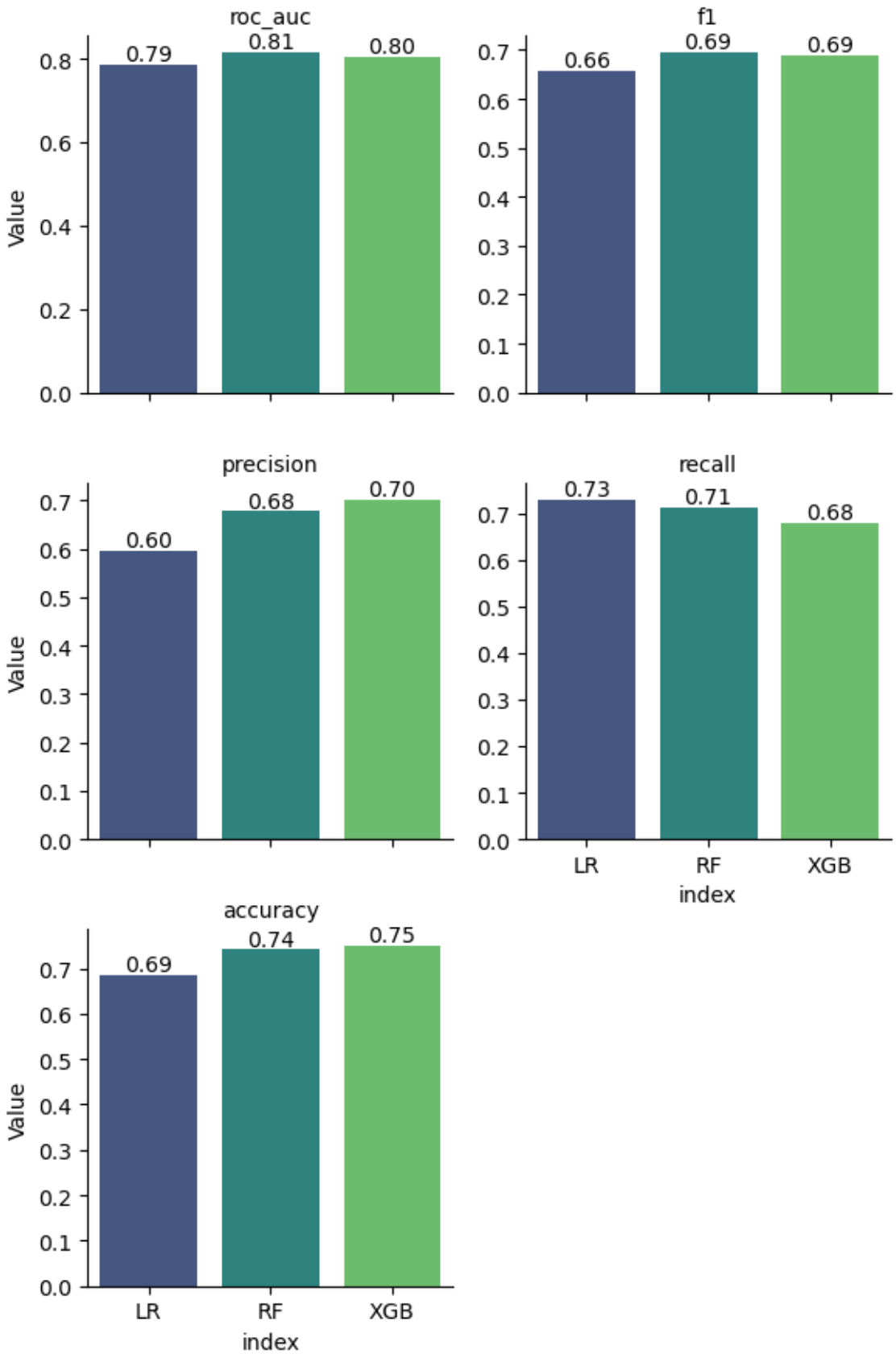

Рисунок 51 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 51, явно отстаёт Логистическая регрессия. Несмотря на то, что её Reall достаточно высок, она обладает худшим балансом среди всех моделей, а также худшей классифицирующей способностью.

Random Forrest и XGBoost в целом сопоставимые модели в рамках данной задачи. Выбираем Random Forrest, так как незначительно выше Recall и ROC-AUC.

Выводы

Задача классификации CC_{50} была решена.

Были проведены следующие этапы:

5. Предобработка данных, получение целевой переменной, очистка датасета от выбросов.
6. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей.
7. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.

Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками ROC-AUC и Recall

2.3 Классификация SI

Задача классификации SI решалась аналогично задаче классификации IC_{50}.

2.3.1 Предобработка данных

После загрузки данных была сформирована целевая величина (рисунок 52):

```
df["SI_gt_median"] = (df[target] > df[target].median()).astype(int)
```

Рисунок 52 – получение целевой величины

```
class_ratio = np.mean(y_train_ic50)
print(f"Баланс классов: {class_ratio:.2f} / {1-class_ratio:.2f}")
```

```
Баланс классов: 0.51 / 0.49
```

Рисунок 53 – отношение меток классов

Классы распределены поровну.

После этого проведено удаление выбросов. В данном случае использовано правило 3σ. Результаты приведены на рисунке 38

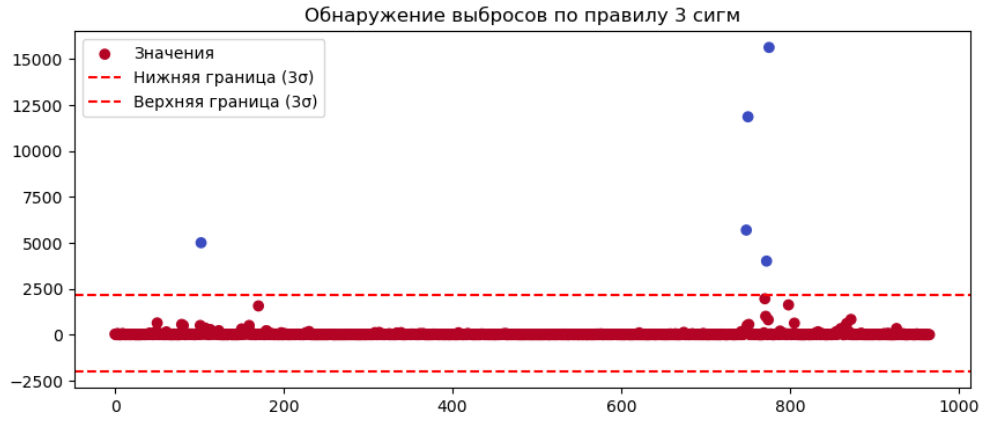

Рисунок 54 – отсечённые выбросы

После отсечения в датасете осталось 961 строка.

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета.

2.3.2 Бейзлайн

В качестве бейзлайна были выбраны модели аналогичные моделям из задачи классификации IC_{50}

Для обучение всех моделей была использована кросс-валидация типа Stratified K-Fold.

Результаты бейзлайна приведены на рисунке 55

	model	cv_mean_roc_auc	cv_std_roc_auc	ROC-AUC	F1	PRECISION	RECALL
0	logreg	0.660159	0.025633	0.724808	0.651515	0.641791	0.661538
1	rf	0.686892	0.031053	0.703942	0.595745	0.552632	0.646154
2	xgb	0.659296	0.030765	0.684231	0.638298	0.592105	0.692308

Рисунок 55 – метрики бейзлайна

Как видно, все модели показывают себя средне.

2.3.3 Оптимизация моделей

Оптимизация проводилась аналогично задаче IC_{50}.

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 56 - 66

```
Final Metrics on Test Set:
roc_auc: 0.7313
f1: 0.6202
precision: 0.6250
recall: 0.6154
accuracy: 0.6621
```

Рисунок 56 – результаты работы лучшей модели логистической регрессии на тестовой выборке

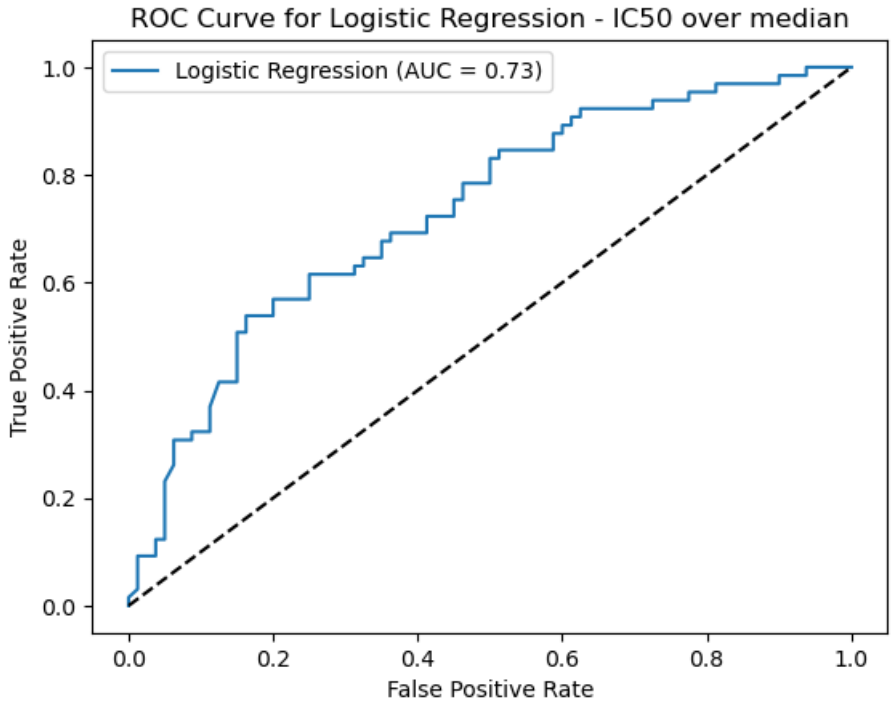

Рисунок 57 – ROC кривая для логистической регрессии

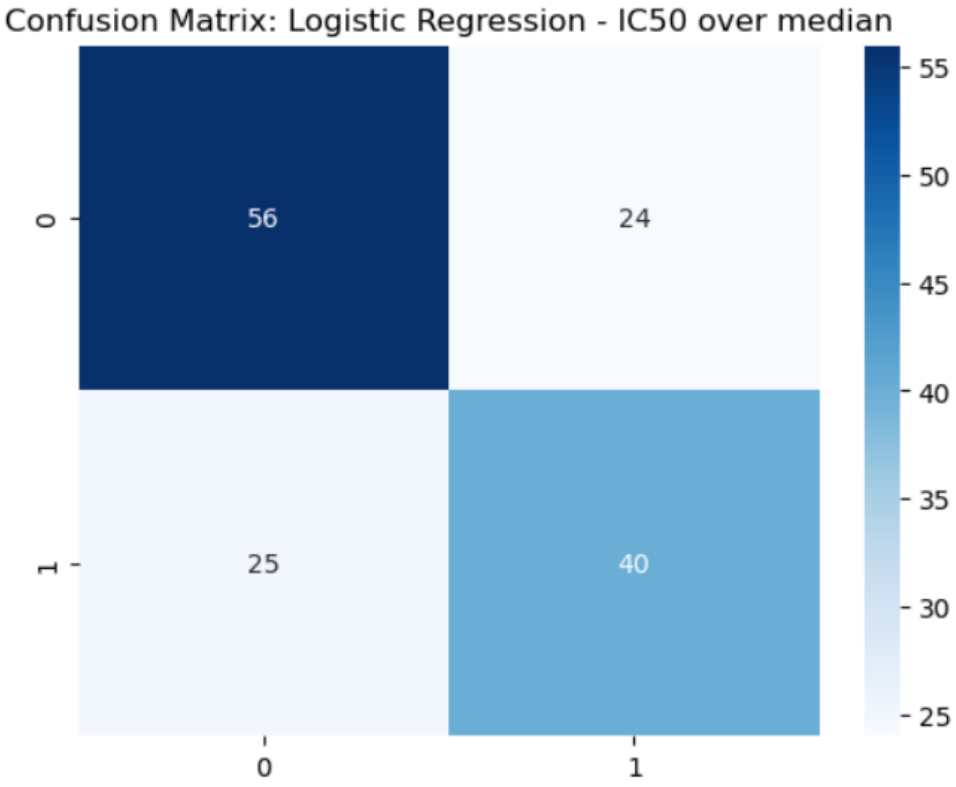

Рисунок 58 – матрица ошибок для логистической регрессии

```
Final Metrics on Test Set:
roc_auc: 0.7267
f1: 0.6015
precision: 0.5882
recall: 0.6154
accuracy: 0.6345
```

Рисунок 59 – результаты работы лучшей модели Random Forest на тестовой выборке

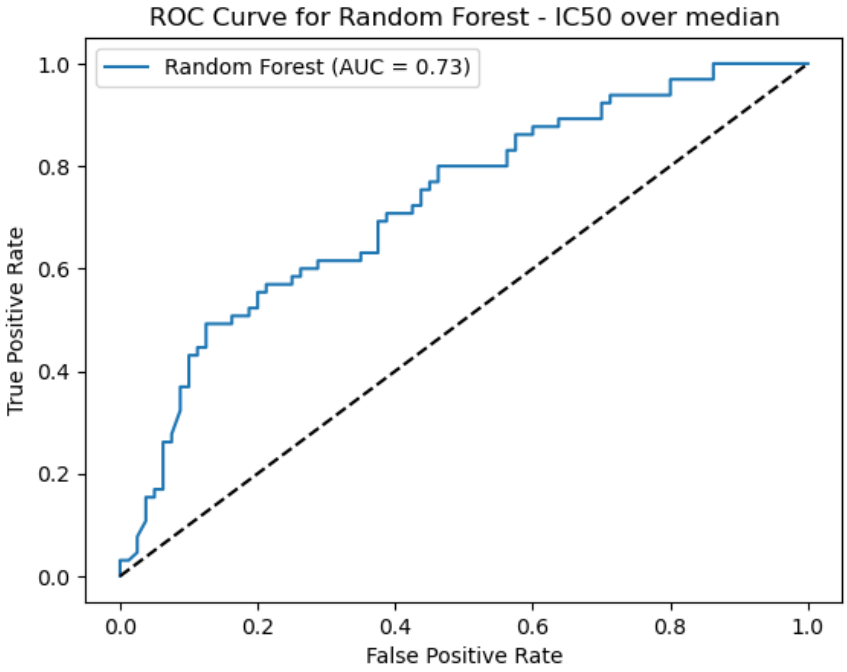

Рисунок 60 – ROC-кривая для Random Forest

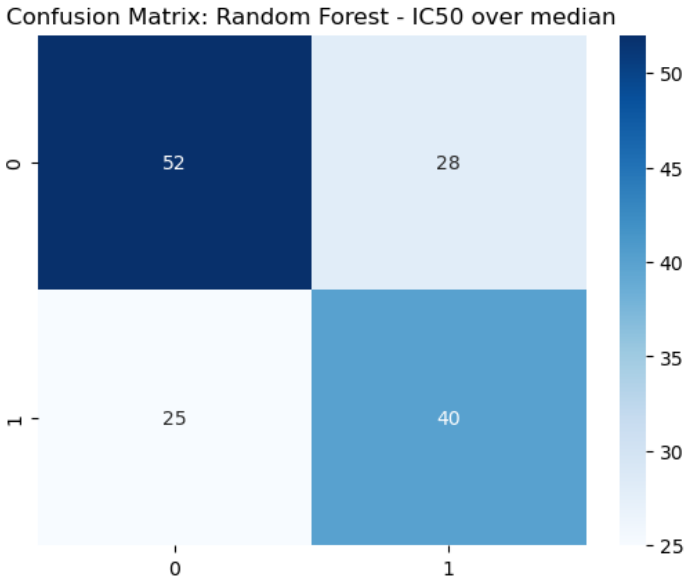

Рисунок 61 – матрица ошибок для Random Forest

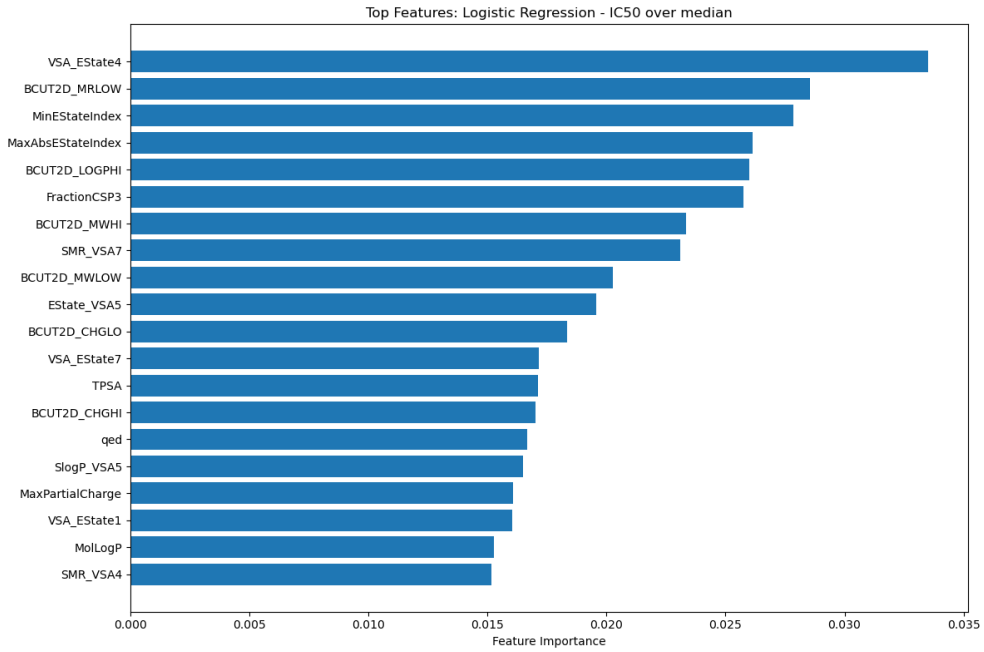

Рисунок 62 – важность признаков для Random Forest

```
Final Metrics on Test Set:
roc_auc: 0.7267
f1: 0.6015
precision: 0.5882
recall: 0.6154
accuracy: 0.6345
```

Рисунок 63 – результаты работы лучшей модели XGBoost на тестовой выборке

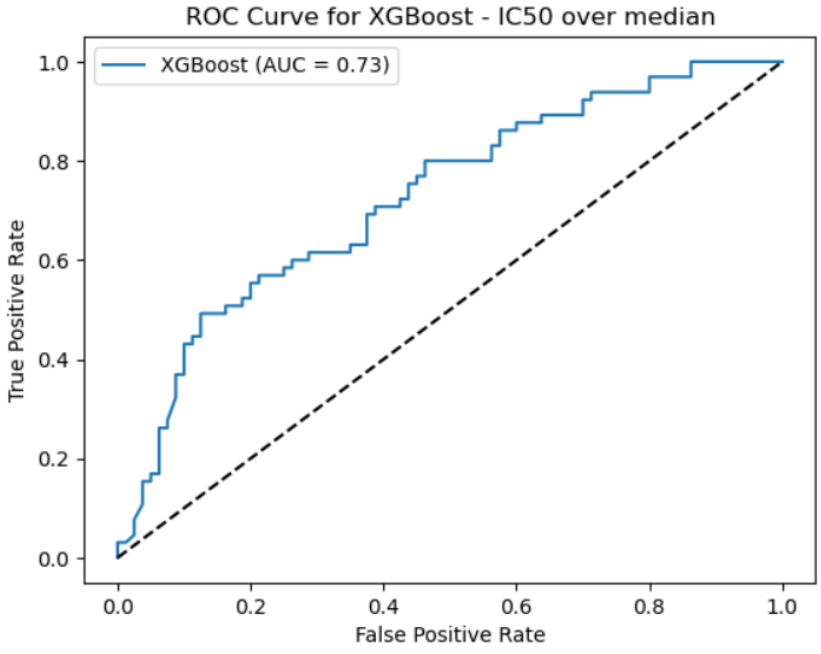

Рисунок 64 – ROC-кривая для XGBoost

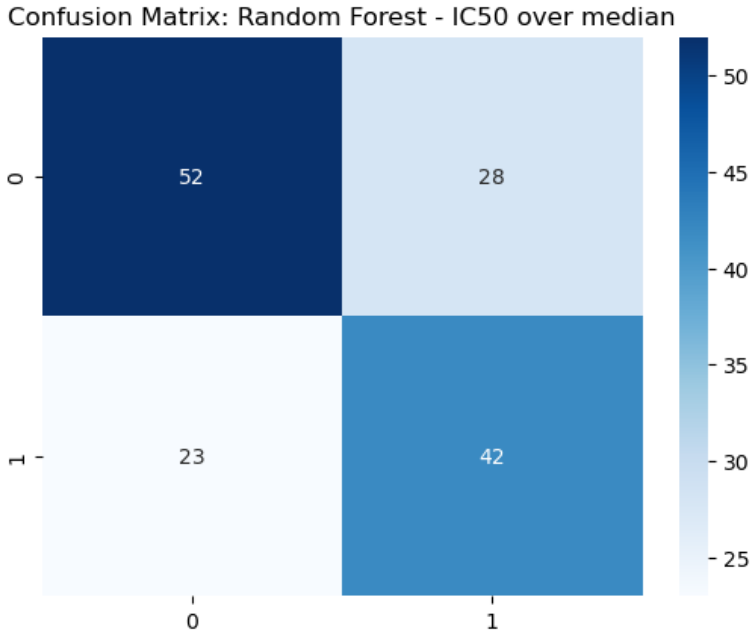

Рисунок 65 – матрица ошибок для XGBoost

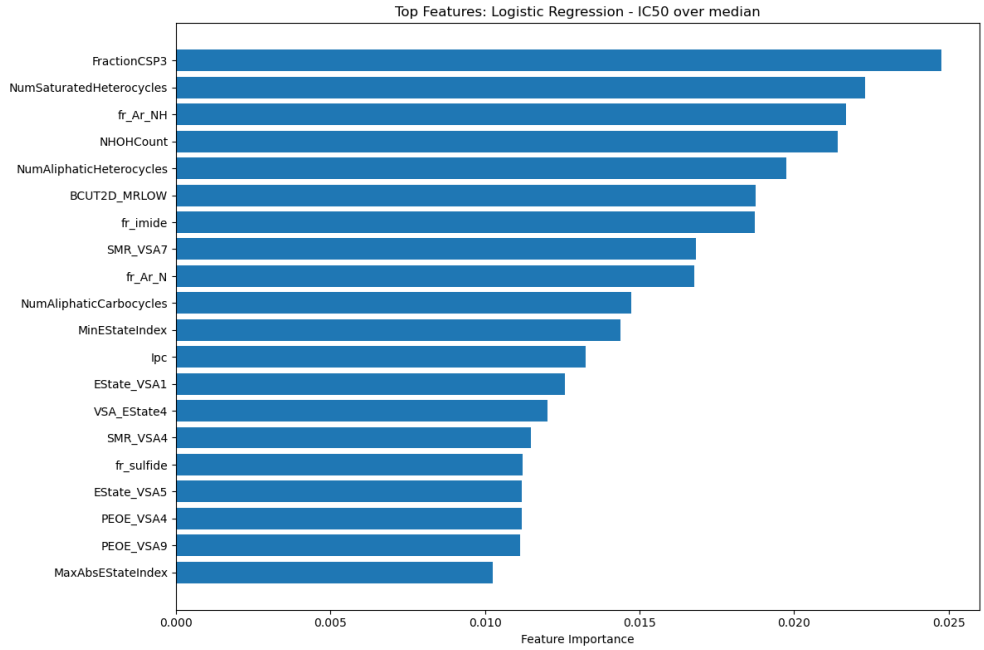

Рисунок 66 – важные признаки для XGBoost

2.3.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики ROC-AUC и Recall

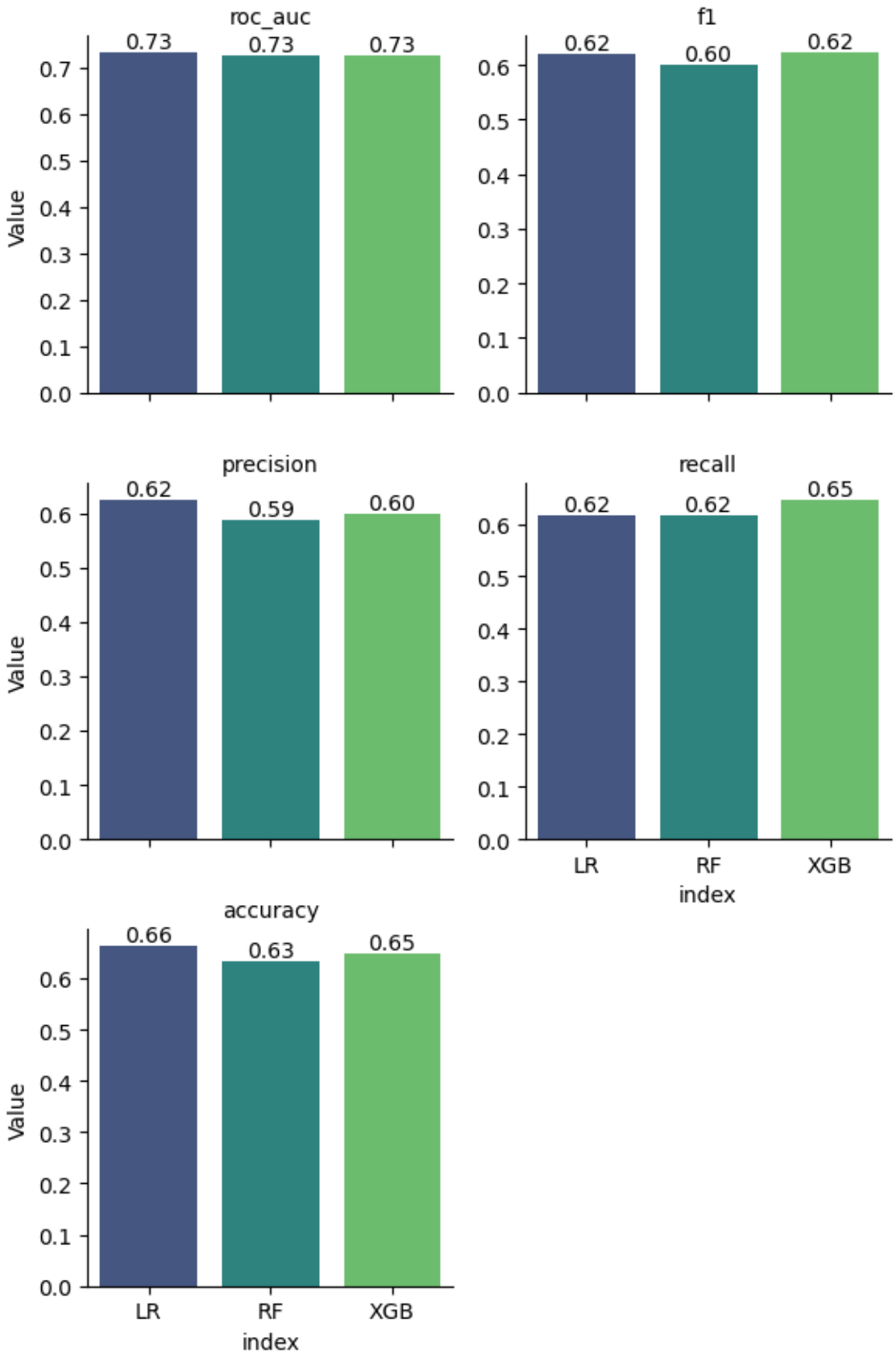

Рисунок 67 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 67, явно отстаёт Случайный Лес: и по балансу, и по точности.

Логистическая Регрессия и XGBoost в целом сопоставимые модели в рамках данной задачи. Выбираем XGBoost, так как незначительно выше Recall

Выводы

Задача классификации SI была решена.

Были проведены следующие этапы:

1. Предобработка данных, получение целевой переменной, очистка датасета от выбросов.
2. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей.
3. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.

Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками ROC-AUC и Recall

2.4 Классификация SI > 8

Задача классификации SI решалась аналогично задаче классификации IC_{50}.

2.4.1 Предобработка данных

После загрузки данных была сформирована целевая величина (рисунок 68):

```
df["SI_gt_8"] = (df[target] > 8).astype(int)
```

Рисунок 68 – получение целевой величины

```
class_ratio = np.mean(y_train_ic50)
print(f"Баланс классов: {class_ratio:.2f} / {1-class_ratio:.2f}")
```

```
Баланс классов: 0.35 / 0.65
```

Рисунок 69 – отношение меток классов

Классы распределены с дисбалансом. Такой дисбаланс не внесёт значительных последствий, следовательно, нет необходимости дополнительно работать с дисбалансом.

Однако, модель логистической регрессии, скорее всего, покажет худшие результаты при таких вводных. Заменим её на LightGBM.

После этого проведено удаление выбросов. В данном случае использовано правило 3σ. Результаты приведены на рисунке 70

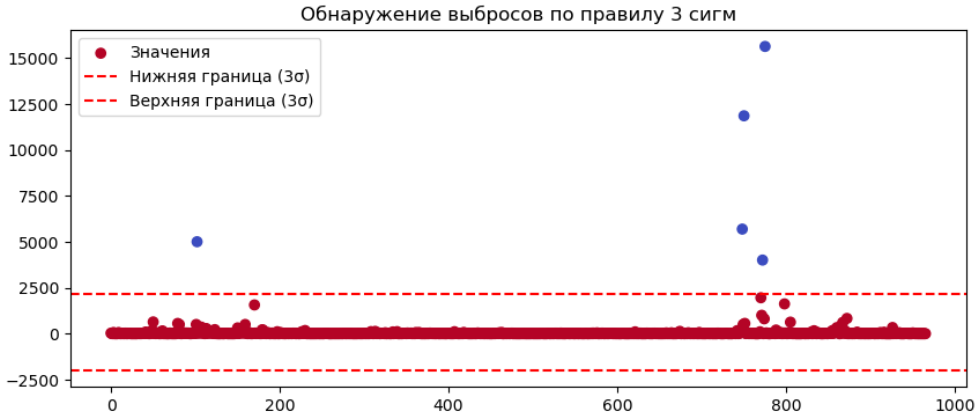

Рисунок 70 – отсечённые выбросы

После отсечения в датасете осталось 961 строка.

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета.

2.4.2 Бейзлайн

В качестве бейзлайна были выбраны модели аналогичные моделям из задачи классификации IC_{50}

Для обучение всех моделей была использована кросс-валидация типа Stratified K-Fold.

Результаты бейзлайна приведены на рисунке 71

	model	cv_mean_roc_auc	cv_std_roc_auc	ROC-AUC	F1	PRECISION	RECALL
0	logreg	0.698776	0.022441	0.670533	0.511111	0.547619	0.479167
1	rf	0.723705	0.030281	0.733570	0.527473	0.558140	0.500000
2	xgb	0.711271	0.026022	0.701031	0.489362	0.500000	0.479167

Рисунок 71 – метрики бейзлайна

Как видно, все модели показывают себя средне.

2.4.3 Оптимизация моделей

Оптимизация проводилась аналогично задаче IC_{50}.

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 72 – 82

```
Final Metrics on Test Set:
roc_auc: 0.7476
f1: 0.5669
precision: 0.4557
recall: 0.7500
accuracy: 0.6207
```

Рисунок 72 – результаты работы лучшей модели LgihtGBM на тестовой выборке

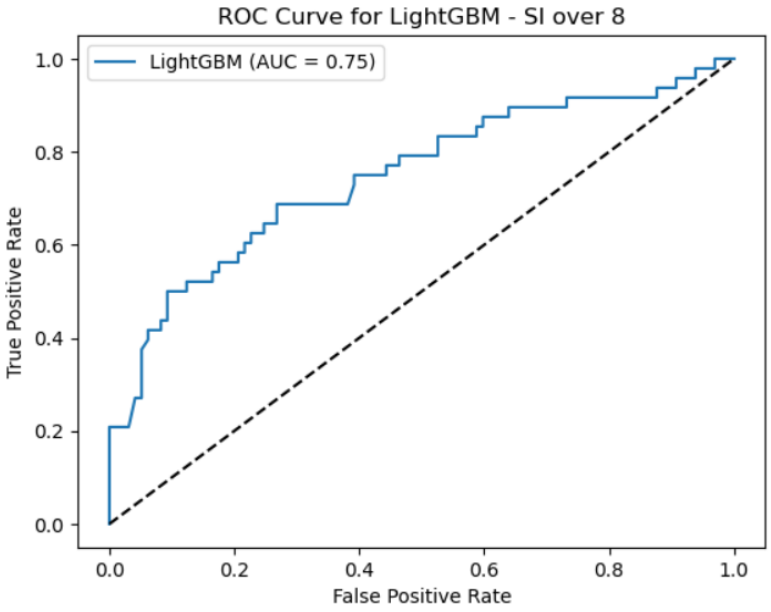

Рисунок 73 – ROC кривая для LgihtGBM

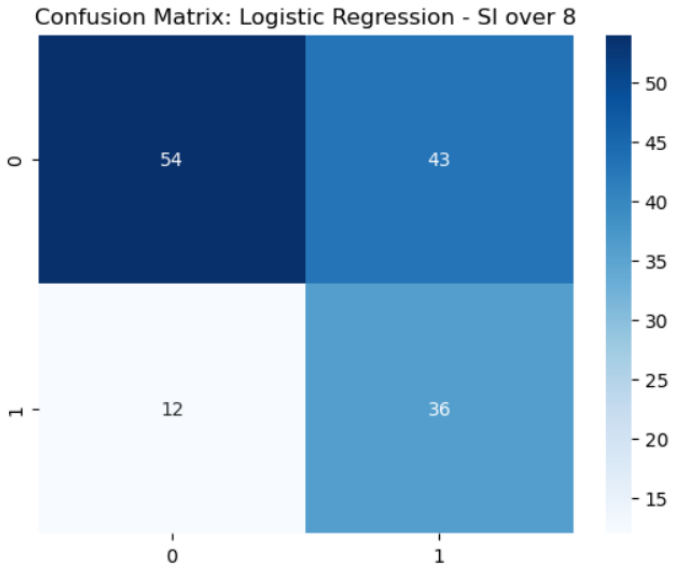

Рисунок 74 – матрица ошибок для LgihtGBM

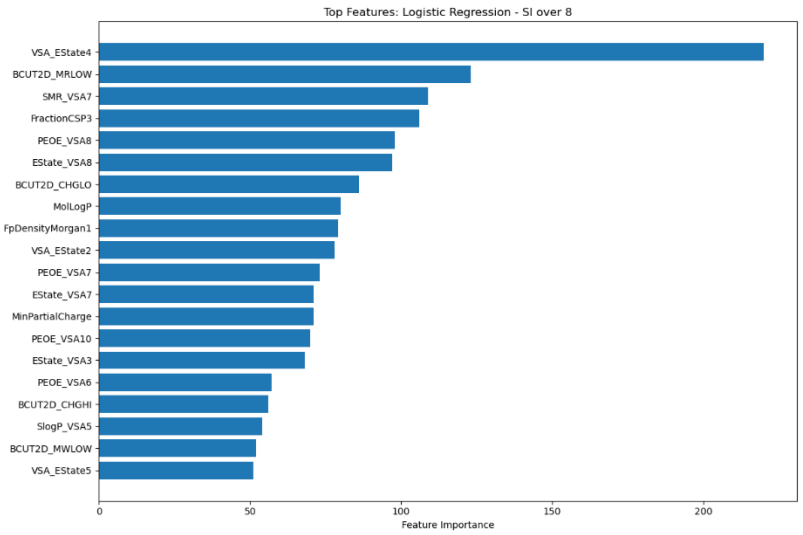

Рисунок 75 – важные признаки для LightGBM

```
Final Metrics on Test Set:
roc_auc: 0.7597
f1: 0.5185
precision: 0.6364
recall: 0.4375
accuracy: 0.7310
```

Рисунок 76 – результаты работы лучшей модели Random Forest на тестовой выборке

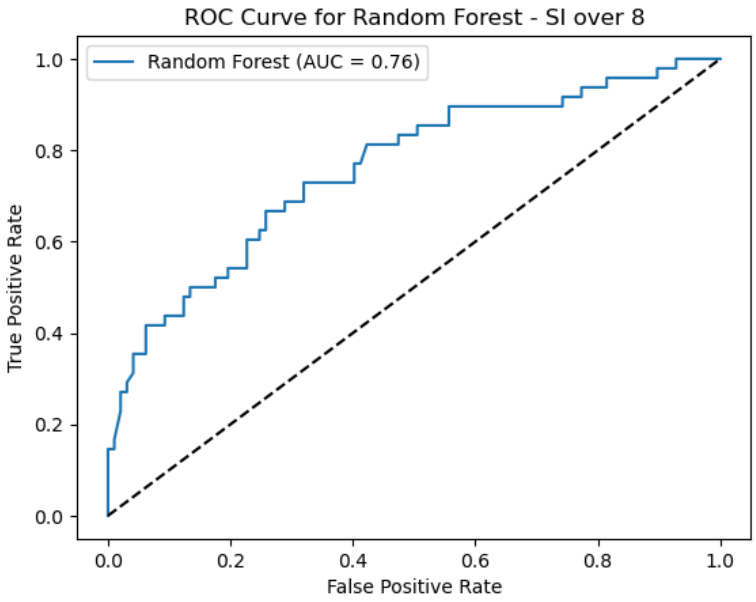

Рисунок 77 – ROC-кривая для Random Forest

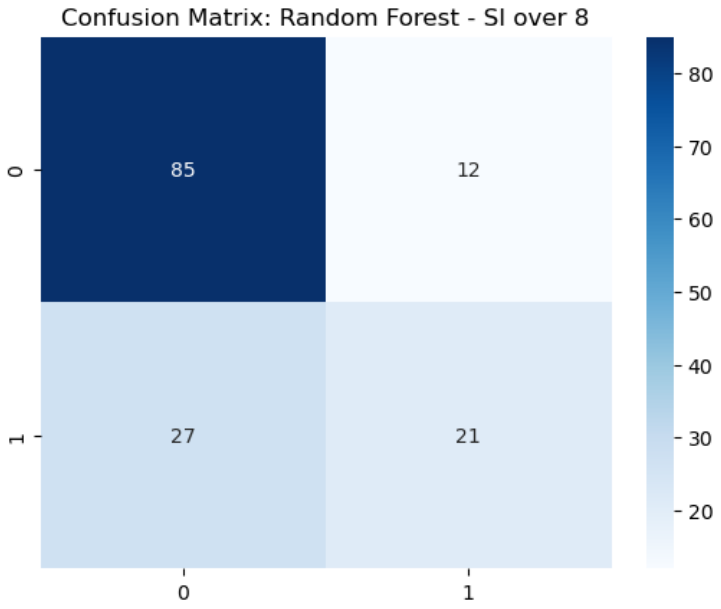

Рисунок 78 – матрица ошибок для Random Forest

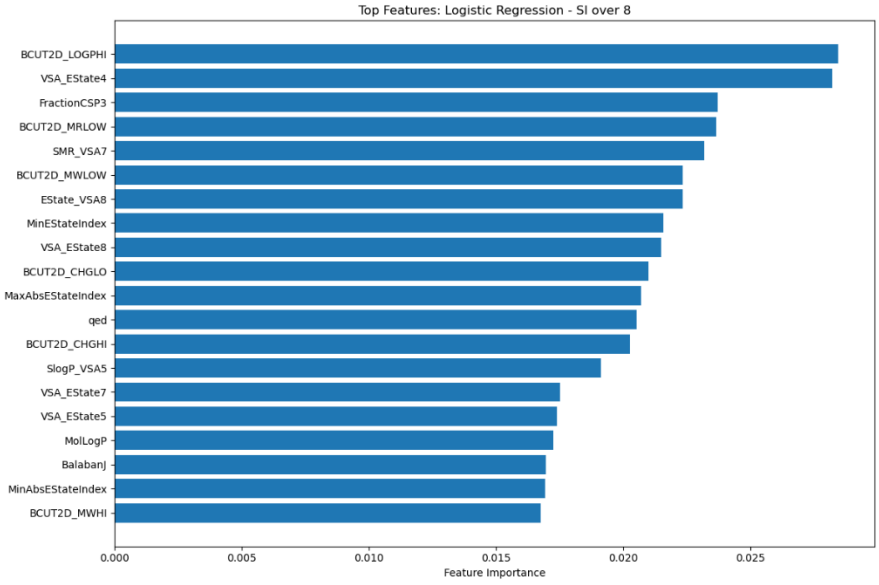

Рисунок 79 – важность признаков для Random Forest

```
Final Metrics on Test Set:
roc_auc: 0.7597
f1: 0.5185
precision: 0.6364
recall: 0.4375
accuracy: 0.7310
```

Рисунок 80 – результаты работы лучшей модели XGBoost на тестовой выборке

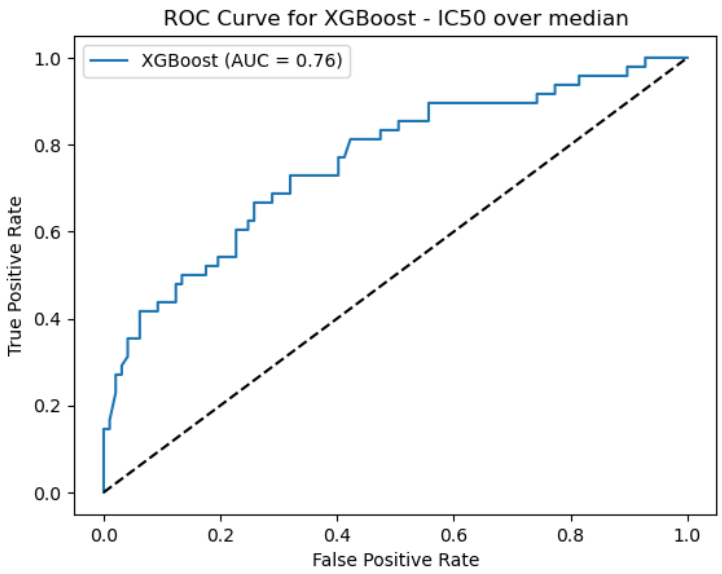

Рисунок 81 – ROC-кривая для XGBoost

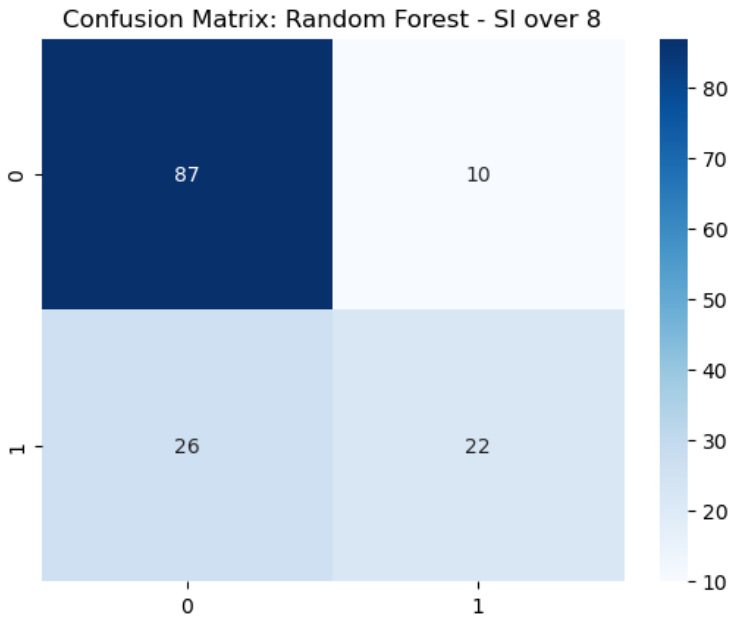

Рисунок 82 – матрица ошибок для XGBoost

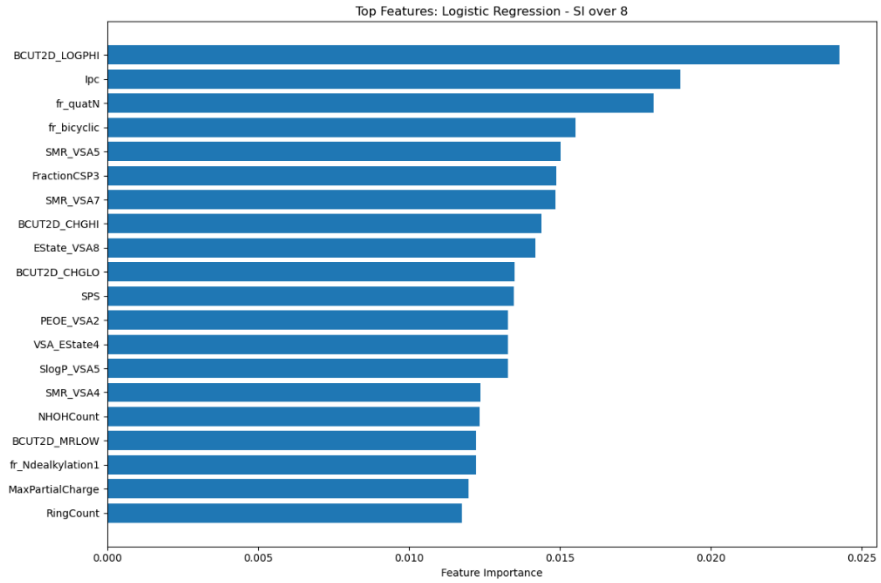

Рисунок 83 – важные признаки для XGBoost

2.4.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики ROC-AUC и Recall

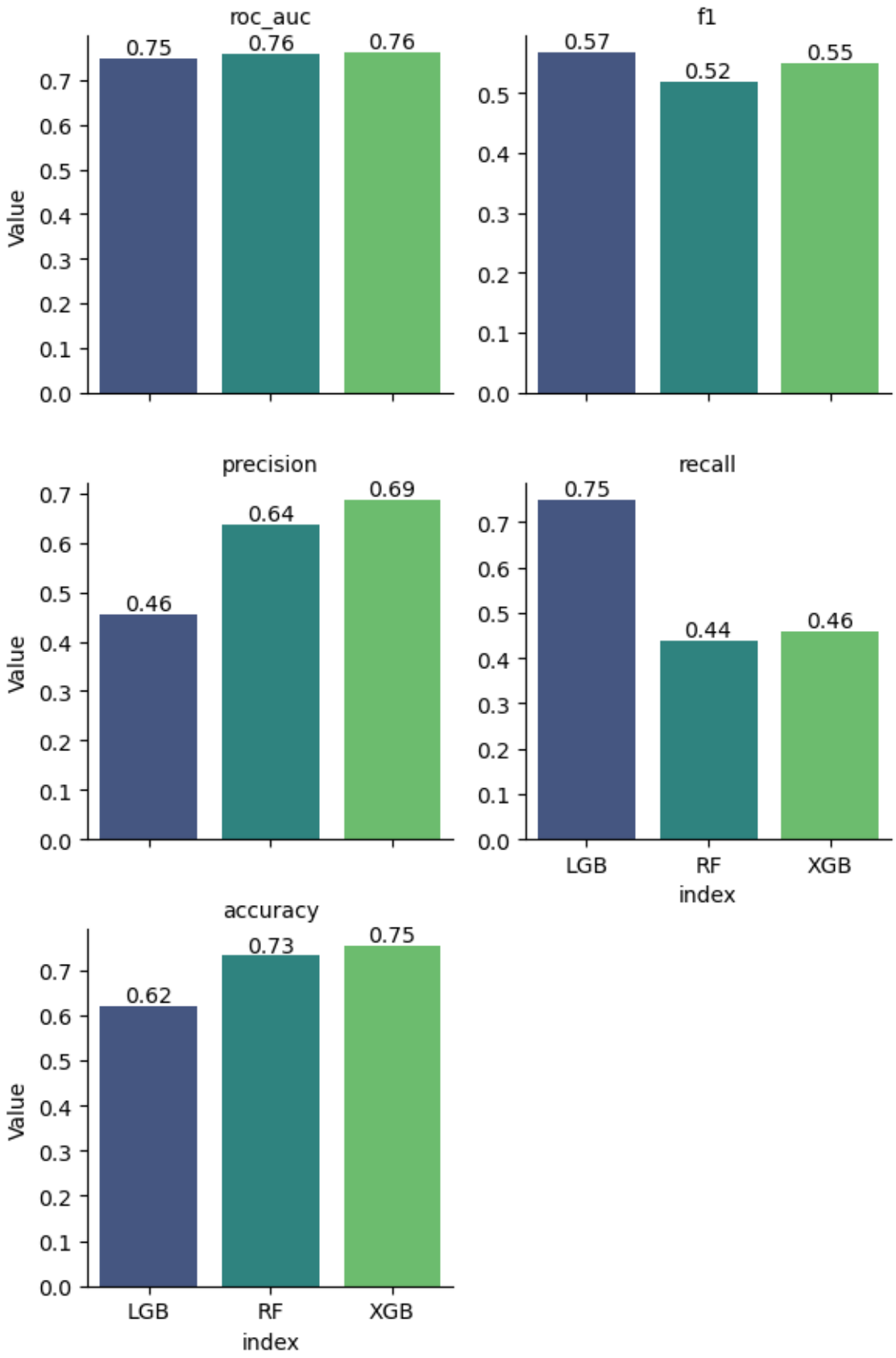

Рисунок 84 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 83, модели принципиально разные. Random Forest отстаёт от XGBoost практически по всем метрикам, поэтому имеет смысл сравнивать только XGBoost и LightGBM.

LightGBM значительно превосходит XGBoost по Recall и незначительно – по балансу, однако проигрывает в точности. Тем не менее, выберем LightGBM, поскольку в рамках данной задачи Recall важнее.

Выводы

Задача классификации SI > 8 была решена.

Были проведены следующие этапы:

4. Предобработка данных, получение целевой переменной, очистка датасета от выбросов.
5. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей.
6. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.

Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками ROC-AUC и Recall

3 Решение задач регрессии

В рамках задачи регрессии необходимо обучить регрессионную модель, способную с достаточной точностью прогнозировать значения целевой величины.

Как правило, регрессионные модели будут показывать худшие значения при решении задач-аналогов для классификации. Например, задача определения того, относится ли соединение к классу активных является более простой относительно задачи прогнозирования значения активности вещества. Стоит отметить, что обе эти задачи могу решаться с одной и той же целью: отделение одних веществ от других (например, активных от неактивных). В таком случае решение регрессионной задачи с высокой долей вероятности не будет являться продуктивным мероприятием, поскольку регрессионная модель будет менее точна в прогнозах.

Однако, есть и другие задачи, связанные с прогнозированием точных значений признаков химических соединений, ввиду чего регрессионную задачу невозможно считать непродуктивной.

Для решения задачи регрессии был разработан следующий план:

1. Преобразование целевой величины (для IC_{50} и CC_{50})
2. Удаление выбросов;
3. Составление обучающей и тестовой выборок;
4. Обучение и базовая проверка бейзлайна;
5. Оптимизации избранных моделей из бейзлайна;
6. Анализ результатов работы моделей;
7. Сравнение моделей и выбор лучшей.

Каждая из реализаций задач соответствует данному плану.

3.1 Регрессионная модель IC50

В рамках данной задачи выполнялась работа по построению регрессионной модели для прогнозирования значений величины IC_{50}

Как было показано в EDA преобразование целевой величины при помощи логарифмирования с использованием десятичного логарифма с высокой вероятностью поможет улучшить результаты работы модели ввиду того, что логарифмирование приблизит характер распределения к нормальному.

3.1.1 Предобработка данных

После загрузки данных было проведено преобразование над целевой величиной (рисунок 85):

```
df["log_10_IC50, mM"] = np.log10(df[target])
```

Рисунок 85 – получение целевой величины

После этого было проведено удаление выбросов. В данном случае использовано правило 3σ.

Рисунок 86 – отсечённые выбросы

После отсечения в датасете осталось 966 строк, что вполне достаточно для обучения без аугментации данных.

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета. Такой объём тестовой выборки был выбран в виду того,

что данных не слишком много и уменьшение объёма обучающей выборки может отрицательно сказаться на обучении модели.

3.1.2 Бейзлайн

В качестве бейзлайна были выбраны следующие модели:

1. Ridge – Линейная модель с L2-регуляризацией. Минимизирует MSE + α * сумма квадратов коэффициентов. Модель устойчива к мультиколлинеарности, работает с высокоразмерными данными, однако не отбирает признаки.
2. Lasso – Линейная модель с L1-регуляризацией. Минимизирует MSE + α * сумма модулей коэффициентов. Модель
3. Random Forest Regressor – Ансамбль деревьев решений с бэггингом. Усредняет предсказания деревьев. Модель работает с нелинейностями и не требует масштабирования данных, однако на малых данных существует риск переобучения.
4. XGBoost Regressore – Градиентный бустинг над деревьями. Оптимизирует функцию потерь с регуляризацией. Модель обладает высокой точностью и способность обрабатывать разреженные данные, однако требует тонкой настройки гиперпараметров для оптимизации результатов. Также модель чувствительна к шуму.
5. SVR – Метод опорных векторов для регрессии. Ищет "трубку" вокруг предсказаний. Модель показывает достаточно высокую эффективность при малых выборках, работает с нелинейностями через ядра, однако является чувствительно к масштабированию данных.

В качестве метрик были выбраны следующие:

1. R^2 – Коэффициент детерминации, показывающий, насколько модель эффективнее решения конкретной задачи путём

прогнозирования всех величин исключительно средним значением по выборке. Будет использоваться для общей оценки качества модели и для сравнения моделей друг с другом.

2. RMSE – Среднеквадратичная ошибка.
3. MAE – Средняя абсолютная ошибка.
4. MAPE – Средняя абсолютная процентная ошибка.

Было принято, что в рамках данной задачи основным критерием оценки являются коэффициент детерминации, а также MAPE.

Для обучение всех моделей была использована кросс-валидация типа K-Fold.

Результаты бейзлайна приведены на рисунке 23

	model	cv_mean_r2	cv_std_r2	R2	RMSE	MAE	MAPE
0	Ridge	-0.058454	0.427081	-0.138148	0.865856	0.621452	0.563406
1	Lasso	-0.008159	0.009725	-0.000524	0.811820	0.643423	0.676246
2	RandomForest	0.336723	0.079230	0.319948	0.669294	0.515436	0.440340
3	XGBoost	0.222123	0.104278	0.233674	0.710482	0.543741	0.468364
4	SVR	0.320125	0.070305	0.328777	0.664936	0.515920	0.468361

Рисунок 87 – метрики бейзлайна

Как видно, линейные модели явно не справляются с задачей: коэффициент детерминации для этих моделей (как средний на кросс-валидации, так и при работе на тестовой выборке) является отрицательным, что указывает на то, что вместо этих моделей было бы эффективнее прогнозировать значения средним по выборке.

Остальные модели показывают себя достойно. Отдельно отметим XGBoost с достаточно низким коэффициентом детерминации, который может быть обусловлен неоптимальными гиперпараметрами.

Также видно, что Random Forrest имеет склонность к переобучению. Это может быть связано с небольшим объёмом выборки.

Было принято решение проводить оптимизации для моделей Random Forrest, XGBoost и SVR при помощи подбора гиперпараметров.

3.1.3 Оптимизация моделей

В рамках оптимизации для трёх типов моделей, фигурирующих в бейзлайне, были подобраны гиперпараметры при помощи Optuna.

Для моделей были установлены следующие параметры:

- Количество итераций поиска (trial):
  - Random Forest, XGBoost: 100
  - SVR: 30
- Количество фолдов кросс-валидации:
  - Random Forest, XGBoost: 10
  - SVR: 5

Меньшие значения по фолдам и итерациям поиска для метода опорных векторов были выбраны в виду того, что обучение такой модели является затратным по времени и ресурсам.

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 88 – 95

```
Final Metrics on Test Set:
R2: 0.3961
RMSE: 0.6307
MAE: 0.4978
MAPE: 0.4382
```

Рисунок 88 – результаты работы лучшей модели Random Forest на тестовой выборке

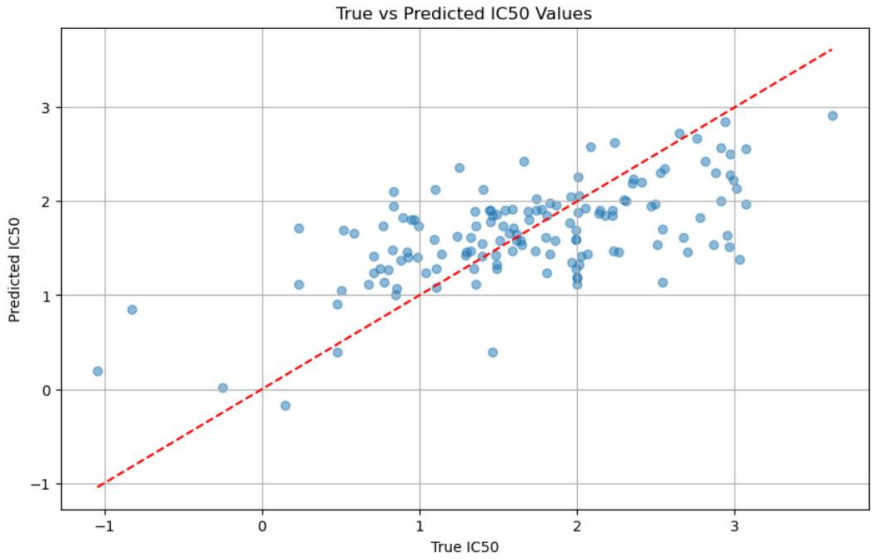

Рисунок 89 – график True vs Predicted для Random Forest

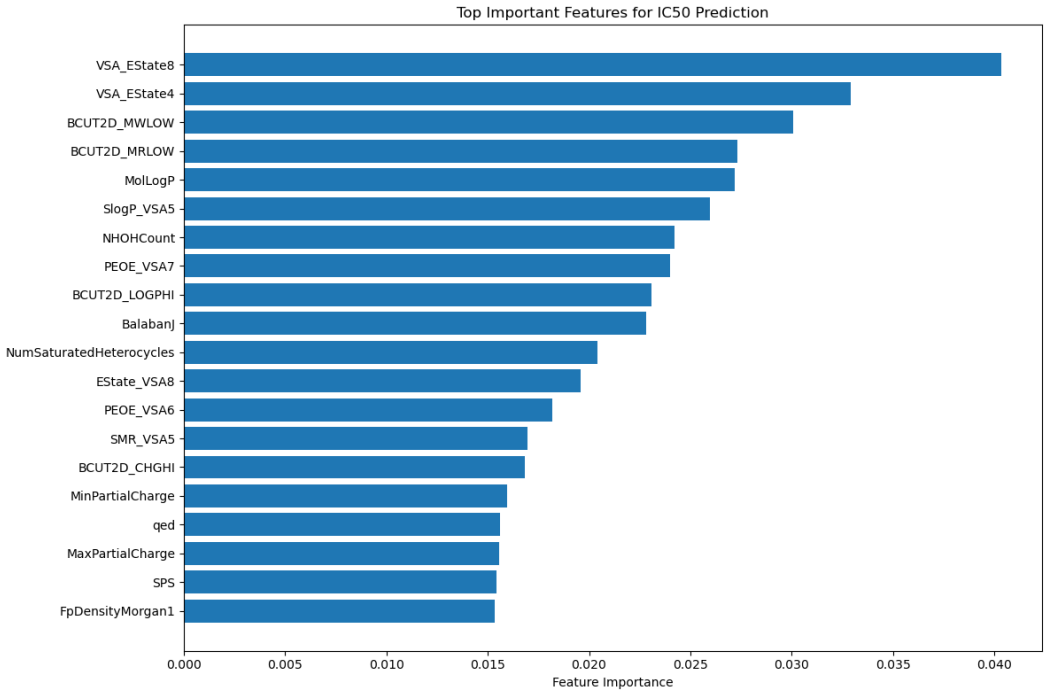

Рисунок 90 – важность признаков для Random Forest

```
Final Metrics on Test Set:
R2: 0.3499
RMSE: 0.6544
MAE: 0.5199
MAPE: 0.4449
```

Рисунок 91 – результаты работы лучшей модели XGBoost на тестовой выборке

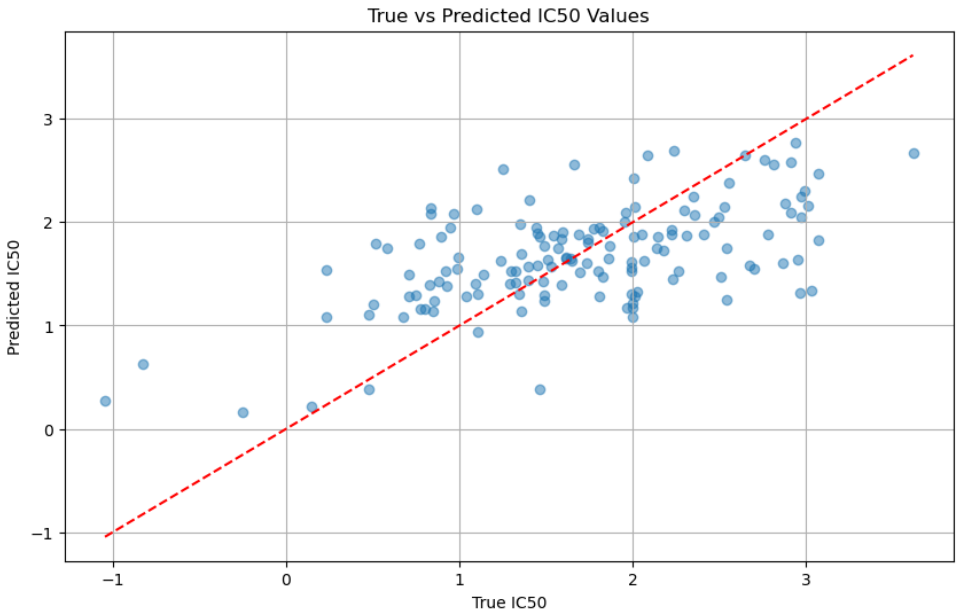

Рисунок 92 – график True vs Predicted для XGBoost

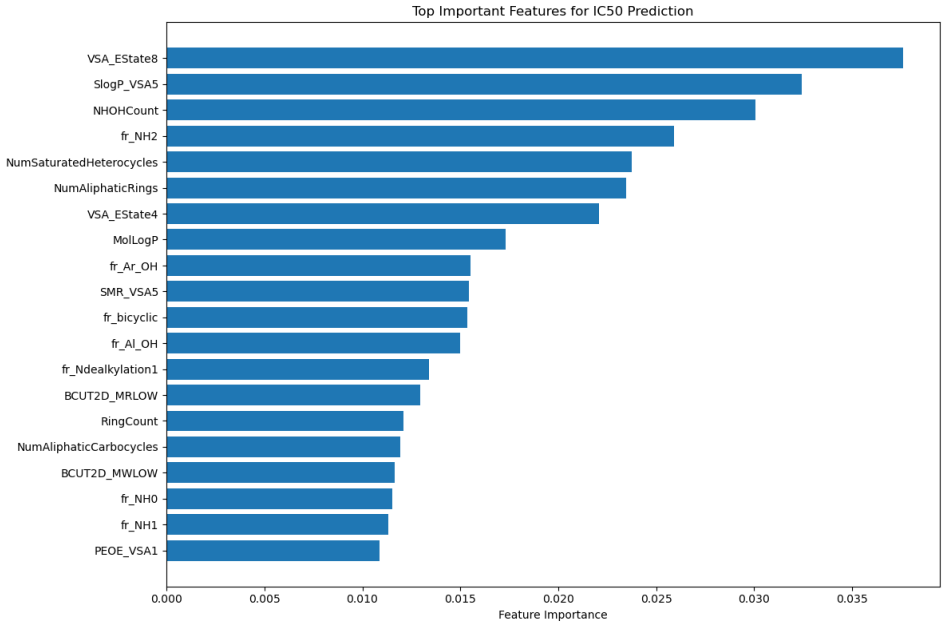

Рисунок 93 – важность признаков для XGBoost

```
Final Metrics on Test Set:
R2: 0.3236
RMSE: 0.6675
MAE: 0.5376
MAPE: 0.4605
```

Рисунок 94 – результаты работы лучшей модели SVR на тестовой выборке

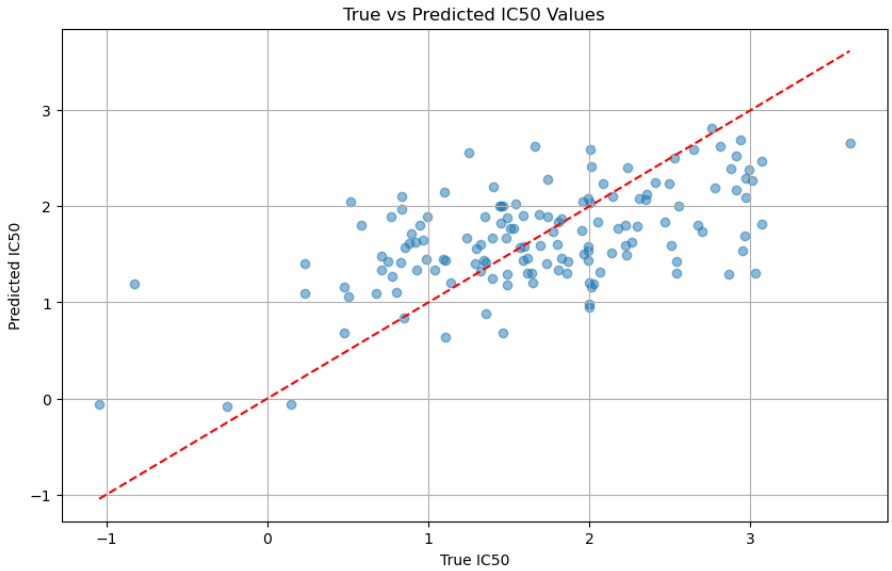

Рисунок 95 – график True vs Predicted для SVR

3.1.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики R^2 и MAPE.

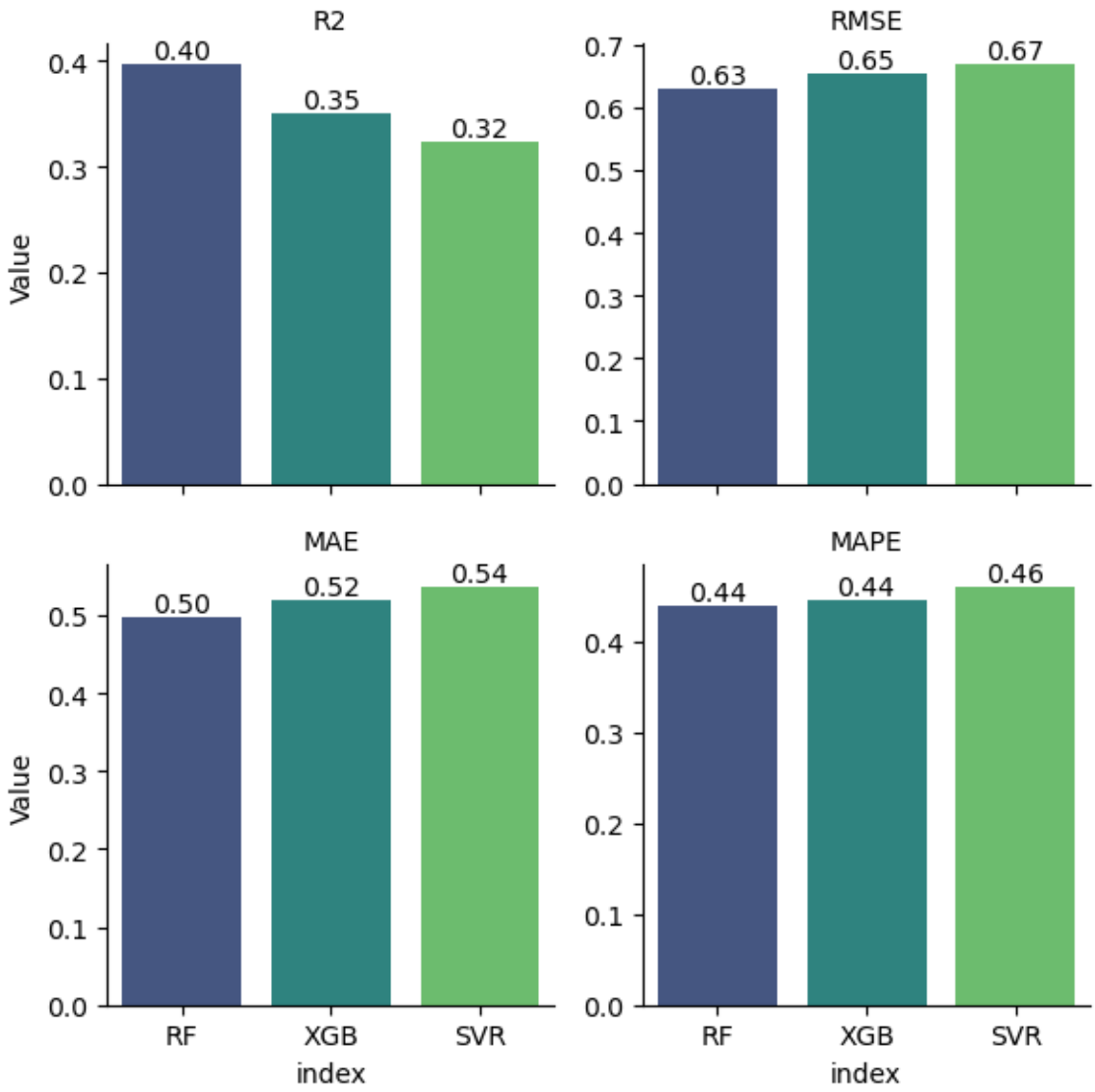

Рисунок 96 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 96, по всем метрикам наилучшие значения показывает Random Forest. Коэффициент детерминации $R^2 = 0.40$ показывает, что модель может объяснить 40% вариации значения целевой переменной, что является недостаточным показателем для утверждения о точности прогноза.

Также стоит обратить внимание на среднюю абсолютную процентную ошибку, значение которой MAPE = 44%. Это достаточно высокая ошибка, которая может значительно влиять на результаты.

Однако для решения задачи классификации веществ (раздел 2) такие показатели гипотетически являются достаточными.

Следовательно, в рамках данной задачи Random Forest является предпочтительным вариантом.

Выводы

Задача регрессии IC_{50} была решена.

Были проведены следующие этапы:

1. Предобработка данных, преобразование целевой переменной, очистка датасета от выбросов.
2. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей. Выбраны модели, которые показали адекватные результаты в рамках бейзлайна.
3. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.

Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками R^2 и MAPE.

Модель Random Forest:

- $R^2 = 0.40$;
- MAPE = 0,44.

3.2 Регрессионная модель CC50

Задача регрессии для CC_{50} решалась аналогично задаче регрессии для IC_{50}.

3.2.1 Предобработка данных

После загрузки данных было проведено преобразование над целевой величиной (рисунок 97):

```
df["log_10_CC50, mM"] = np.log10(df[target])
```

Рисунок 97 – получение целевой величины

После этого было проведено удаление выбросов. В данном случае использовано правило 3σ.

Рисунок 98 – отсечённые выбросы

После отсечения в датасете осталось 950 строк, что вполне достаточно для обучения без аугментации данных.

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета. Такой объём тестовой выборки был выбран в виду того, что данных не слишком много и уменьшение объёма обучающей выборки может отрицательно сказаться на обучении модели.

3.2.2 Бейзлайн

В качестве бейзлайна было принято решене выбрать те же модели, что были выбраны при решении задачи регрессии IC_{50}

Было принято, что в рамках данной задачи основным критерием оценки являются коэффициент детерминации, а также MAPE.

Для обучение всех моделей была использована кросс-валидация типа K-Fold.

Результаты бейзлайна приведены на рисунке 99

	model	cv_mean_r2	cv_std_r2	R2	RMSE	MAE	MAPE
0	Ridge	-0.047528	0.651933	0.164107	0.561113	0.424305	0.235227
1	Lasso	-0.005258	0.003626	-0.021249	0.620213	0.508719	0.286448
2	RandomForest	0.471875	0.060530	0.376245	0.484710	0.349922	0.195116
3	XGBoost	0.417705	0.097679	0.306423	0.511119	0.367526	0.199531
4	SVR	0.463652	0.057710	0.338651	0.499103	0.365159	0.204538

Рисунок 99 – метрики бейзлайна

Вывод по бейзлайну аналогичны выводам для решения задачи регрессии для IC_{50} за тем исключением, что Ridge на тестовой выборке показал положительный коэффициент детерминации. Однако это нивелируется высоким отклонением (cv_std_r2), что вкупе с отрицательным средним показателем R^2 на обучающей выборке свидетельствует о случайности такого значения.

3.2.3 Оптимизация моделей

Оптимизация проводилась аналогично оптимизации моделей для решения задачи регрессии IC_{50}

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 100 – 107.

```
Final Metrics on Test Set:
R2: 0.4006
RMSE: 0.4752
MAE: 0.3501
MAPE: 0.1944
```

Рисунок 100 – результаты работы лучшей модели Random Forest на тестовой выборке

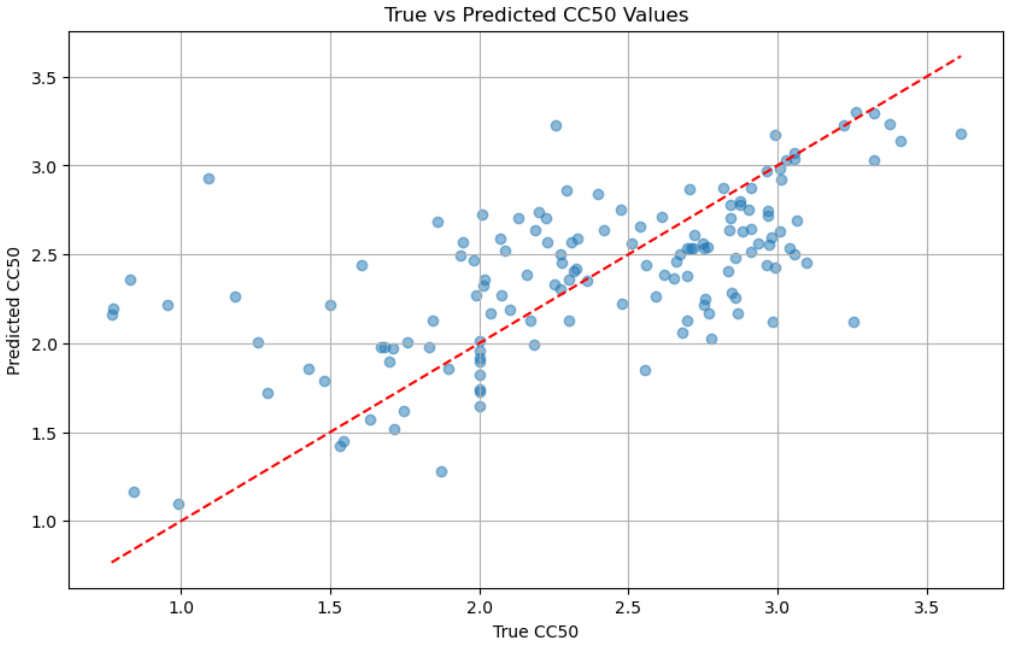

Рисунок 101 – график True vs Predicted для Random Forest

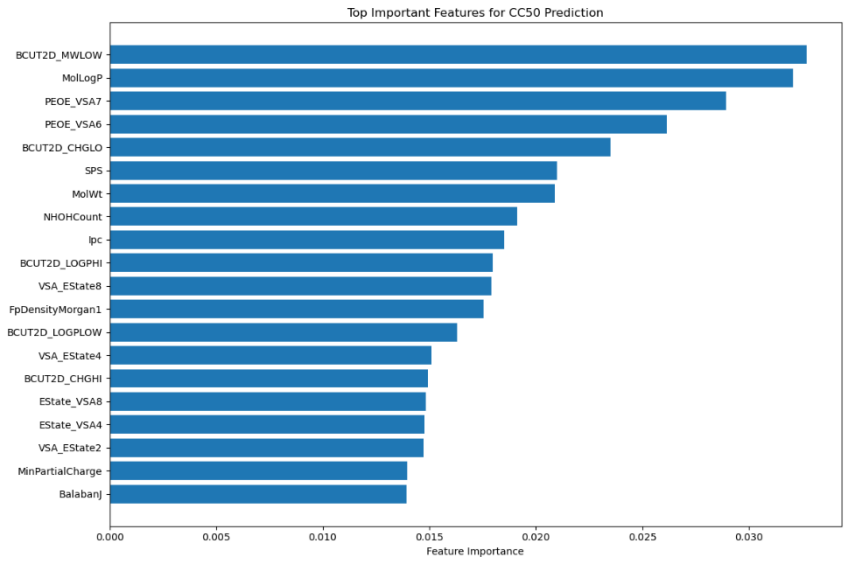

Рисунок 102 – важность признаков для Random Forest

```
Final Metrics on Test Set:
R2: 0.3315
RMSE: 0.5018
MAE: 0.3691
MAPE: 0.2070
```

Рисунок 103 – результаты работы лучшей модели XGBoost на тестовой выборке

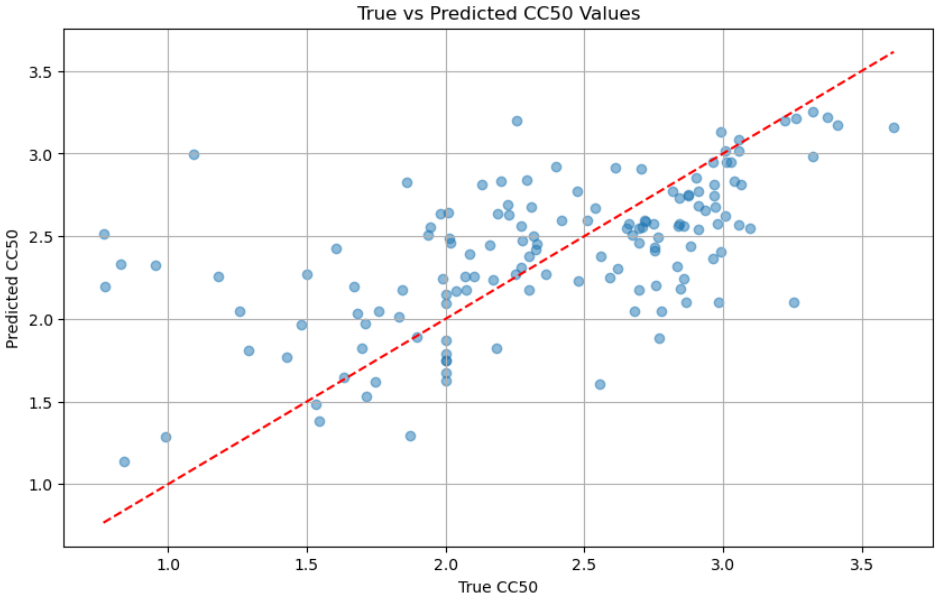

Рисунок 104 – график True vs Predicted для XGBoost

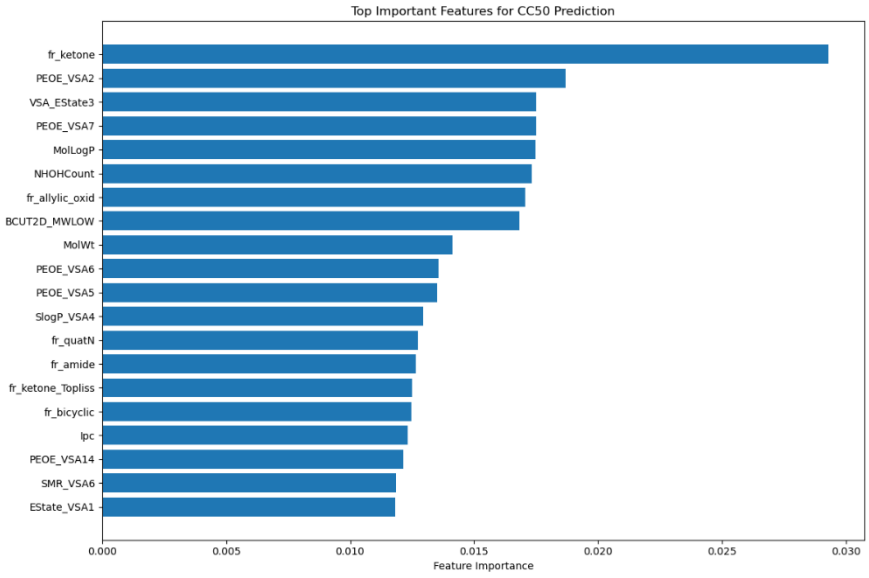

Рисунок 105 – важность признаков для XGBoost

```
Final Metrics on Test Set:
R2: 0.3711
RMSE: 0.4867
MAE: 0.3597
MAPE: 0.1968
```

Рисунок 106 – результаты работы лучшей модели SVR на тестовой выборке

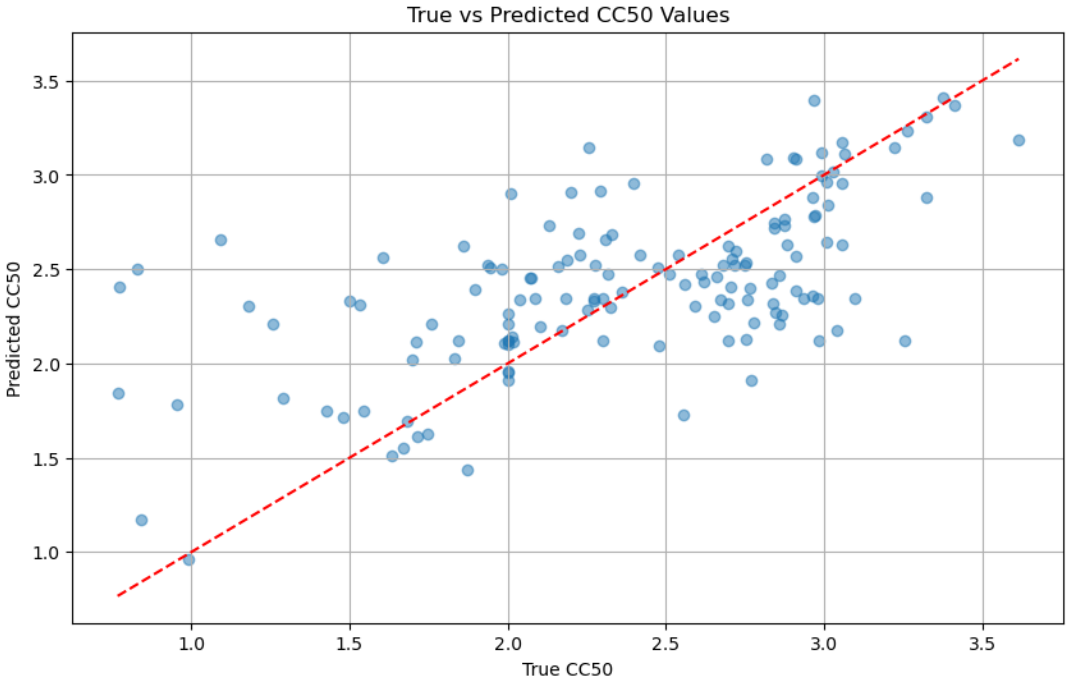

Рисунок 107 – график True vs Predicted для SVR

3.2.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики R^2 и MAPE.

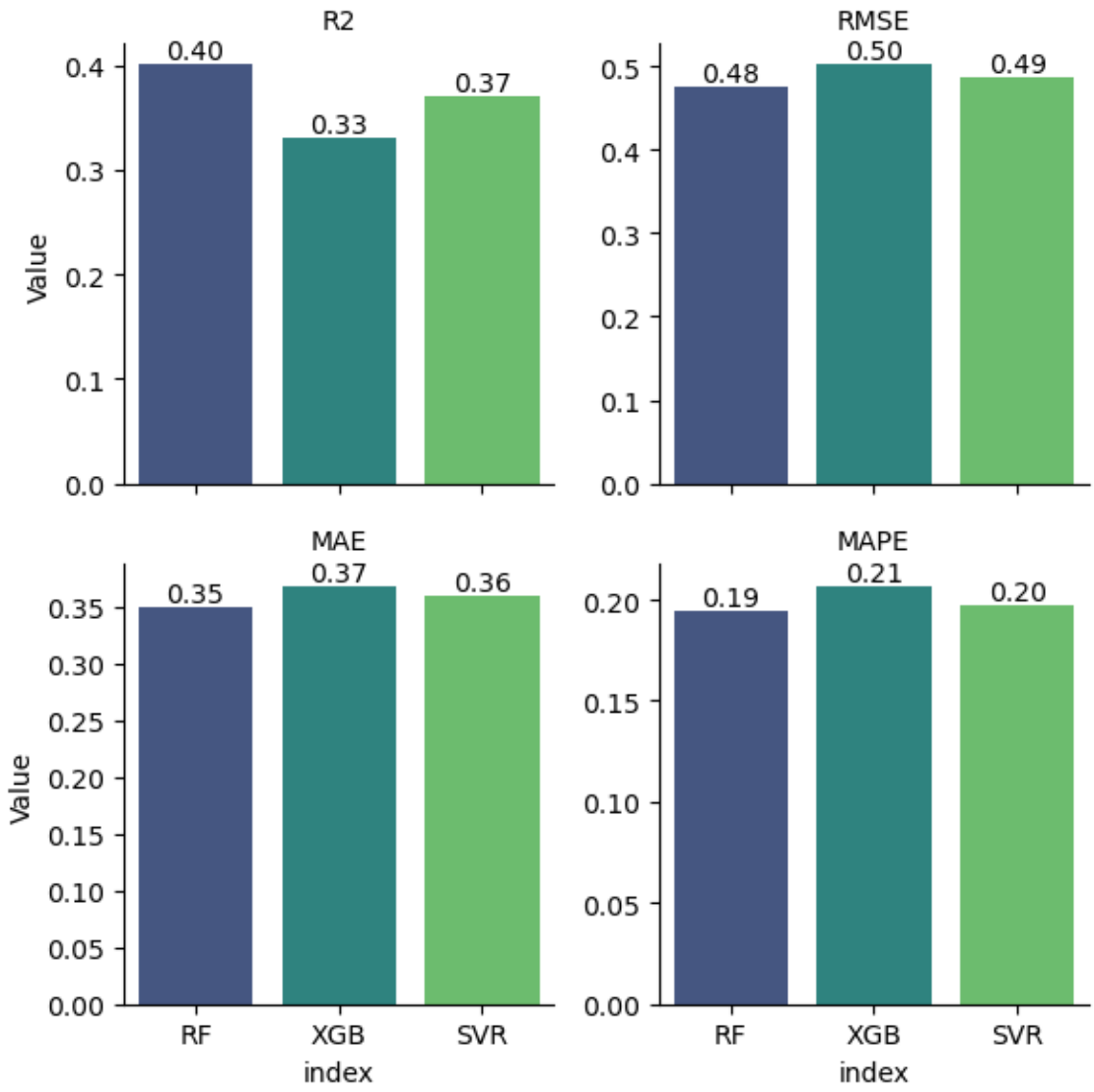

Рисунок 108 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 96, по всем метрикам наилучшие значения показывает Random Forest. Коэффициент детерминации $R^2 = 0.40$ показывает, что модель может объяснить 40% вариации значения целевой переменной, что является недостаточным показателем для утверждения о точности прогноза.

Также стоит обратить внимание на среднюю абсолютную процентную ошибку, значение которой MAPE = 19%. Это значительно более низкая ошибка, чем та, что была получена в задаче регрессии IC_{50}.

Для решения задачи классификации веществ (раздел 2) такие показатели гипотетически являются достаточными.

Следовательно, в рамках данной задачи Random Forest является предпочтительным вариантом.

Выводы

Задача регрессии IC_{50} была решена.

Были проведены следующие этапы:

1. Предобработка данных, преобразование целевой переменной, очистка датасета от выбросов.
2. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей. Выбраны модели, которые показали адекватные результаты в рамках бейзлайна.
3. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.

Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками R^2 и MAPE.

Модель Random Forest:

- $R^2 = 0.40$;
- MAPE = 0,19.

3.3 Регрессионная модель SI

Задача регрессии для SI решалась аналогично задаче регрессии для IC_{50} за тем исключением, что логарифмирование целевой величины не производится, что указано в EDA.

3.3.1 Предобработка данных

После загрузки было проведено удаление выбросов. В данном случае использовано правило межквартильного размаха, поскольку по нему получается добится очситки от большего количества необъяснимых значений, спровоцированных выбросами со стороны двух других целевых величин.

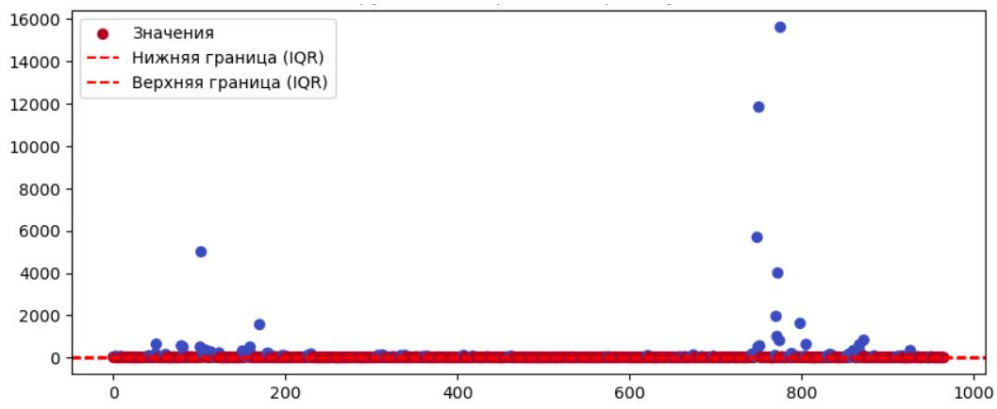

Рисунок 109 – отсечённые выбросы по правилу межквартильного размаха (Q_1 и Q_3)

После отсечения в датасете осталось 847 строк.

Отметим, что логарифмирование после отсечения выбросов выглядит так, как показано на рисунке 110.

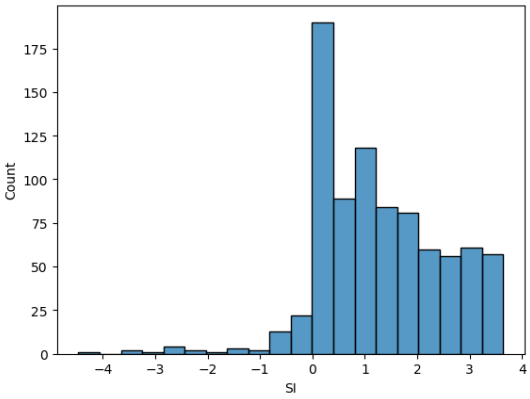

Рисунок 110 – распределение lg(SI)

Как видно из рисунка 110, распределение далеко от нормального, ввиду чего можно сделать вывод о неприменимости логарифмирования в целях улучшения результатов работы моделей.

Данные были разделены на train и test выборки, где test составила 15% от объёма датасета. Такой объём тестовой выборки был выбран в виду того, что данных не слишком много и уменьшение объёма обучающей выборки может отрицательно сказаться на обучении модели.

3.3.2 Бейзлайн

В качестве бейзлайна было принято решене выбрать те же модели, что были выбраны при решении задачи регрессии IC_{50}

Было принято, что в рамках данной задачи основным критерием оценки являются коэффициент детерминации, а также MAPE.

Для обучение всех моделей была использована кросс-валидация типа K-Fold.

Результаты бейзлайна приведены на рисунке 111

	model	cv_mean_r2	cv_std_r2	R2	RMSE	MAE	MAPE
0	Ridge	-0.086966	0.137880	0.228403	89.794231	64.572138	2.053788
1	Lasso	0.127103	0.083317	0.270788	87.293129	60.282802	1.876895
2	RandomForest	0.204572	0.108726	0.380450	80.462094	49.670423	1.411732
3	XGBoost	0.051447	0.135166	0.272932	87.164673	52.859318	1.365244
4	SVR	-0.024932	0.065603	-0.069738	105.728424	59.488465	1.005160

Рисунок 111 – метрики бейзлайна

Вывод по бейзлайну аналогичны выводам для решения задачи регрессии для IC_{50}, за исключением того, что Lasso и Ridge показали хорошие результаты на тестовой выборке, а SVR справился значительно хуже.

3.3.3 Оптимизация моделей

Оптимизация проводилась аналогично оптимизации моделей для решения задачи регрессии IC_{50}

После подбора гиперпараметров были получены результаты, которые представлены на рисунках 100 – 107.

```
Final Metrics on Test Set:
R2: 0.2788
RMSE: 86.8131
MAE: 58.2149
MAPE: 2.5048
```

Рисунок 112 – результаты работы лучшей модели Random Forest на тестовой выборке

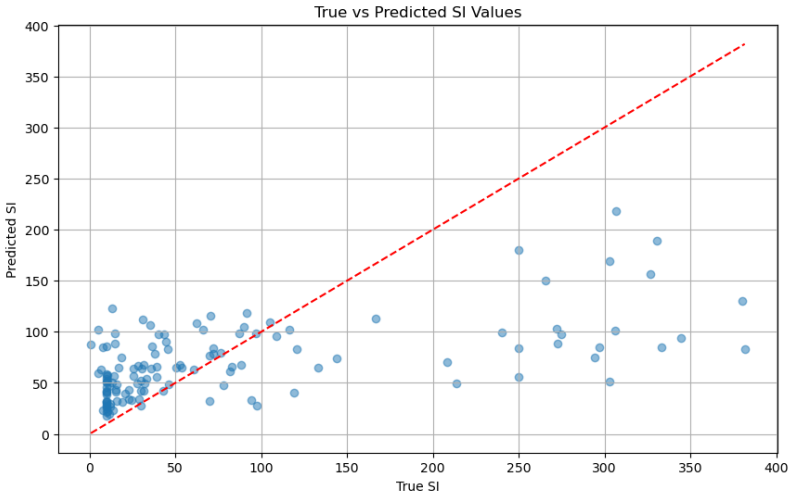

Рисунок 113 – график True vs Predicted для Random Forest

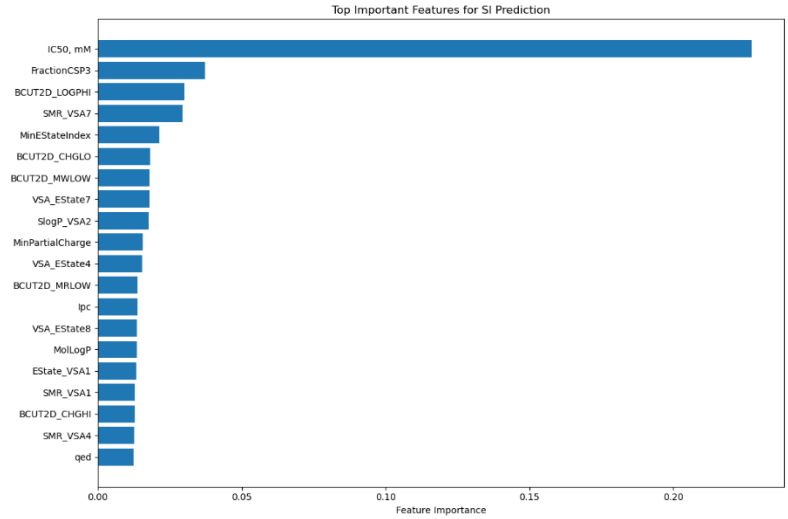

Рисунок 114 – важность признаков для Random Forest

```
Final Metrics on Test Set:
R2: 0.3702
RMSE: 81.1254
MAE: 50.3477
MAPE: 1.5746
```

Рисунок 115 – результаты работы лучшей модели XGBoost на тестовой выборке

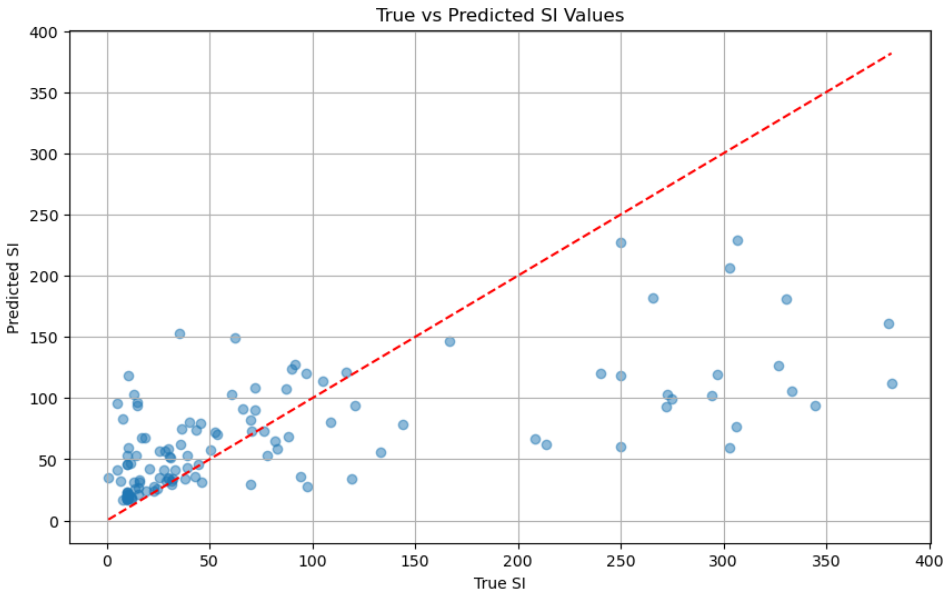

Рисунок 116 – график True vs Predicted для XGBoost

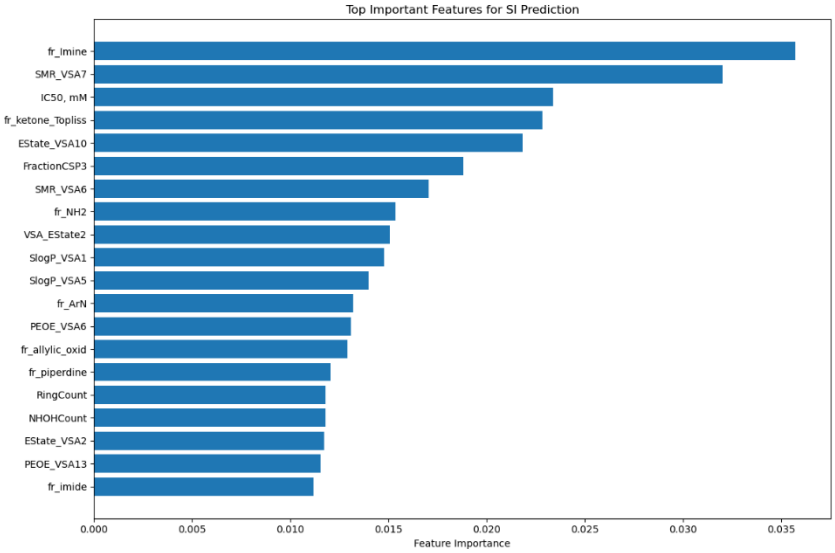

Рисунок 117 – важность признаков для XGBoost

```
Final Metrics on Test Set:
R2: 0.2163
RMSE: 90.4937
MAE: 54.3287
MAPE: 1.5887
```

Рисунок 118 – результаты работы лучшей модели SVR на тестовой выборке

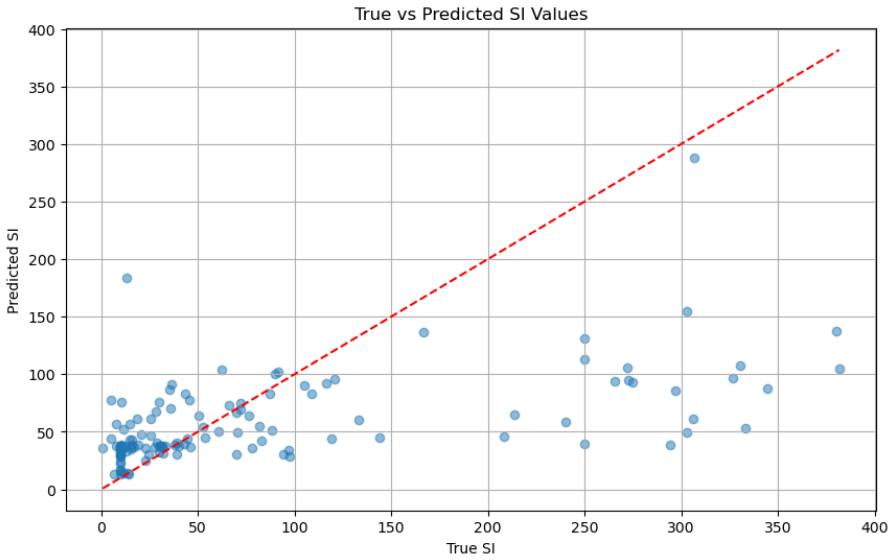

Рисунок 119 – график True vs Predicted для SVR

3.3.4 Сравнение результатов моделей

Сравнение моделей будем проводить, ориентируясь на метрики R^2 и MAPE.

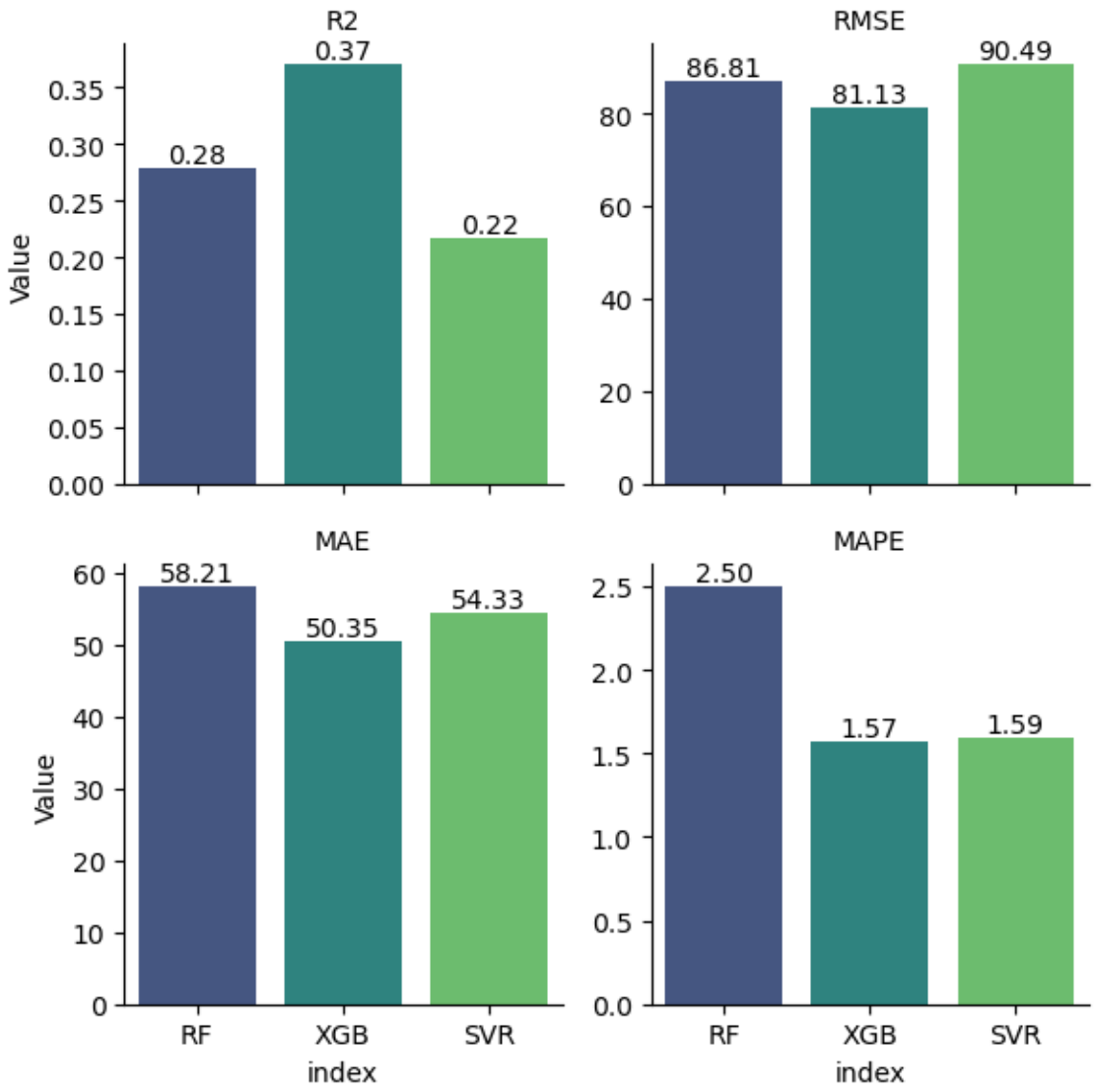

Рисунок 120 – метрики моделей при работе на тестовой выборке

Как видно на рисунке 96, по всем метрикам наилучшие значения показывает Random Forest. Коэффициент детерминации $R^2 = 0.40$ показывает, что модель может объяснить 40% вариации значения целевой переменной, что является недостаточным показателем для утверждения о точности прогноза.

Также стоит обратить внимание на среднюю абсолютную процентную ошибку, значение которой MAPE = 19%. Это значительно более низкая ошибка, чем та, что была получена в задаче регрессии IC_{50}.

Для решения задачи классификации веществ (раздел 2) такие показатели гипотетически являются достаточными.

Следовательно, в рамках данной задачи Random Forest является предпочтительным вариантом.

Выводы

Задача регрессии IC_{50} была решена.

Были проведены следующие этапы:

4. Предобработка данных, преобразование целевой переменной, очистка датасета от выбросов.
5. Подготовлены модели и метрики для проведения испытаний. Обучены бейзлайны моделей. Выбраны модели, которые показали адекватные результаты в рамках бейзлайна.
6. Проведена оптимизация моделей при помощи кросс-валидации и подбора оптимальных гиперпараметров на базе Optuna.

Была выбрана модель, которая удовлетворяла критериям выбора: обладала самыми высокими метриками R^2 и MAPE.

Модель Random Forest:

- $R^2 = 0.40$;
- MAPE = 0,19.

Заключение

В ходе выполнения курсовой работы решены поставленные задачи по построению и оптимизации моделей машинного обучения для прогнозирования и классификации активности (IC_{50}), токсичности (CC_{50}) и селективности (SI) лекарственных соединений. Ключевые достижения и выводы:

1. Результаты EDA и предобработки данных

Очистка данных: Удалены дубликаты (32 записи) и строки с пропусками (0.3% данных), итоговый объём выборки — 966 соединений.

Анализ распределений:

IC_{50} и CC_{50} имеют левоасимметричное распределение с выбросами. Логарифмирование ($\log_{10}$) улучшило их свойства для регрессионных задач.

SI сохранил концентрацию значений около нуля; логарифмирование признано нецелесообразным.

Корреляционный анализ:

Выявлена умеренная корреляция между IC_{50} и CC_{50} (коэффициент Пирсона = 0.52), подтверждающая связь активности и токсичности.

Удалены 154 пары признаков с высокой корреляцией (коэффициент Пирсона > 0,9) и 19 признаков с нулевой дисперсией.

Нормализация: Применен StandardScaler для обеспечения стабильности моделей.

2. Результаты классификации

Для бинарной классификации использованы ансамблевые методы. Ключевые метрики — ROC-AUC и Recall (важно минимизировать пропуск активных/токсичных соединений):

Результаты моделей по задачам классификации приведены в таблице

Таблица 1 – результаты моделей для задач классификации

Задача	Лучшая модель	ROC-AUC	Recall	Особенности
IC_{50} > медианы	Random Forest	0.77	0.63	Сбалансированность Precision-Recall
CC_{50} > медианы	Random Forest	0.81	0.71	Высокие общие результаты
SI > медианы	XGBoost	0.73	0.65	Лучшие метрики
SI > 8	LightGBM	0.75	0.75	Лучший Recall

В задачах IC_{50}/CC_{50} Random Forest показал максимальную устойчивость.

Для SI > 8 (дисбаланс классов) LightGBM превзошёл XGBoost по Recall благодаря эффективной работе с редкими классами.

3. Результаты регрессии

Для прогнозирования непрерывных значений использованы преобразования (логарифмирование IC_{50}/CC_{50}) и ансамблевые методы. Основные метрики: R^2 и MAPE:

Результаты работы регрессионных моделей приведены в таблице 2.

Таблица 2 – результаты регрессионных моделей

Задача регрессии	Лучшая модель	R^2	MAPE	Интерпретация
IC_{50}	Random Forest	0.40	44%	Объясняет 40% дисперсии; высокая ошибка
CC_{50}	Random Forest	0.40	19%	наименьшая ошибка
SI	Random Forest	0.40	19%	Наименьшая ошибка

Проблемы:

- Низкий R^2 (0.4) указывает на сложность прогнозирования точных значений, что может достигаться из-за зашумлённости данных, малого объёма выборки
- Линейные модели (Ridge, Lasso) показали неадекватные результаты ($R^2 < 0$).

4. Общие выводы

Эффективность методов:

1. Ансамблевые алгоритмы (Random Forest, XGBoost, LightGBM) превзошли линейные модели благодаря способности улавливать нелинейные зависимости.
2. Для классификации предпочтительны Random Forest (сбалансированные классы) и LightGBM (дисбаланс). В регрессии доминирует Random Forest.

Качество прогнозов:

1. Классификаторы достигли средне-высоких ROC-AUC (0.73–0.81), что позволяет использовать их для ранней фильтрации соединений.
2. Регрессионные модели имеют ограниченную точность ($R^2 = 0.4$), но могут применяться для ранжирования соединений по активности/токсичности.

5. Рекомендации по повышению качества работы моделей

Улучшение данных:

1. Предоставить более объёмный набор данных;
2. Апробировать преобразования над некоторыми признаками.

Оптимизация моделей:

- Тестировать стекинг ансамблей для регрессии.
- Применить Feature Engineering: преобразование выбросов, генерация полиномиальных признаков.

Итог.

Построенные модели позволяют ранжировать соединения по активности и токсичност. Классификаторы демонстрируют более высокие результаты, регрессионные подходы требуют доработки для повышения точности.

Работа подтверждает перспективность методов машинного обучения в хемоинформатике.