

"Прогноз эффективности противовирусных соединений с использованием методов машинного обучения"

Проблематика

В современной фармацевтической индустрии разработка новых препаратов – сложный, многоступенчатый процесс, который требует значительных временных, интеллектуальных и финансовых ресурсов. Дополнительным вызовом является необходимость комплексного тестирования препаратов на различных этапах разработки.

Временные затраты - классический процесс разработки лекарственного препарата может занимать от 10 до 15 лет, что существенно замедляет появление жизненно важных медикаментов на рынке.

Высокая стоимость исследований - экспериментальное тестирование тысяч химических соединений требует колоссальных финансовых вложений, при этом большинство кандидатов оказываются неэффективными.

Неопределенность прогнозирования – сложность предсказания эффективности и безопасности соединений в краткосрочной и долгосрочной перспективе без проведения дорогостоящих экспериментов.

Учитывая данную проблематику, применение методов машинного обучения на различных этапах разработки лекарственных соединений может существенно дополнить традиционные методы и улучшить результаты. Это достигается за счет синергии подходов, замены части дорогостоящих реальных исследований математическим прогнозированием, эффективной обработки и систематизации больших объемов информации, а также минимизации влияния субъективного восприятия исследователя на процесс отбора перспективных соединений.

Постановка задачи

Для анализа предоставлены данные о 1000 химических соединениях с указанием их эффективности против вируса гриппа. Параметры, характеризующие эффективность, обозначаются как IC50 (концентрация полумаксимального ингибирования), CC50 (концентрация полумаксимальной цитотоксичности) и SI (индекс селективности).

На основании предоставленных данных от химиков необходимо построить прогноз, позволяющий подобрать наиболее эффективное сочетание параметров для создания лекарственных препаратов.

Основные задачи:

1. Провести исследовательский анализ данных (EDA)
2. Построить модели регрессии для прогнозирования IC50, CC50 и SI
3. Построить модели классификации для определения превышения медианных значений IC50, CC50 и SI
4. Построить модель классификации для определения $SI > 8$

Ожидаемые результаты:

EDA – провести и описать комплексный анализ данных, выявить значимые признаки и закономерности, на их основе подготовить данные для эффективного решения поставленных задач.

Решение задач регрессии и классификации – сравнить различные модели с настройкой гиперпараметров, сделать выводы о применимости методов, дать рекомендации по улучшению качества прогнозирования и практическому применению результатов.

Предварительная обработка данных

Датасет (размерностью 1001x214) содержит данные по 1001 химическому соединению, каждая строка соответствует определенному веществу, 3 столбца – целевые характеристики вещества, остальные столбцы – его характеристики (структурные, химические, физические и молекулярные).

Unnamed: 0	IC50, mM	CC50, mM	SI	MaxAbsEStateIndex	MaxEStateIndex	MinAbsEStateIndex	MinEStateIndex	qed	SP5	...	fr_sulfide	fr_sulfonamid	fr_sulfone	
0	0	6.239374	175.482382	28.125000	5.094096	5.094096	0.387225	0.387225	0.417362	42.928571	...	0	0	0
1	1	0.771831	5.402819	7.000000	3.961417	3.961417	0.533868	0.533868	0.462473	45.214286	...	0	0	0
2	2	223.806778	161.142320	0.720000	2.627117	2.627117	0.543231	0.543231	0.268923	42.187500	...	0	0	0
3	3	1.705624	107.855854	63.235294	5.097360	5.097360	0.390803	0.390803	0.377846	41.862069	...	0	0	0
4	4	107.131532	138.270991	1.300000	5.150510	5.150510	0.270476	0.270476	0.429038	36.514286	...	0	0	0

5 rows x 214 columns

Выполнена проверка данных на наличие пропусков

Первоначальная форма: (1001, 214)

Форма после удаления пропущенных данных: (998, 214)

Анализ целевых переменных IC50, CC50, SI

IC50 — это концентрация вещества, при которой оно ингибирует 50% активности целевого белка/фермента/процесса. Чем меньше IC50, тем выше активность соединения.

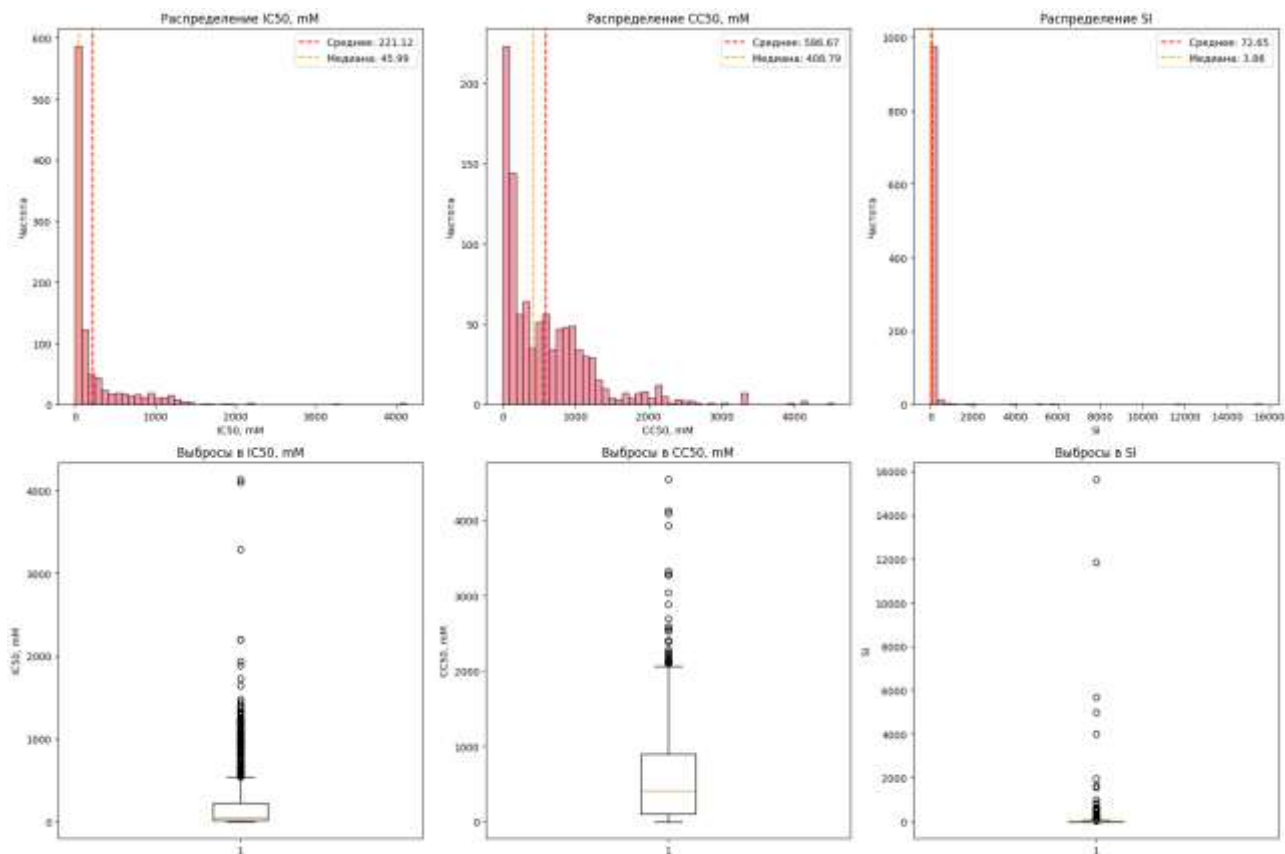
CC50 – это 50% Cytotoxic Concentration — это концентрация вещества, при которой оно вызывает гибель 50% клеток в клеточной культуре. Этот показатель используется для оценки цитотоксичности (токсичности для клеток) химического соединения или лекарства.

SI - Индекс селективности

Для целевых переменных датасета выполняется соотношение $SI = CC50 / IC50$

	IC50, mM	CC50, mM	SI
count	998.000000	998.000000	998.000000
mean	221.118757	586.668414	72.650005
std	400.510657	642.016454	685.504279
min	0.003517	0.700808	0.011489
25%	12.491340	99.999036	1.457233
50%	45.992006	408.793314	3.856410
75%	224.408630	891.770961	16.525000
max	4128.529377	4538.976189	15620.600000





Анализ выбросов данных:

IC50, mM:

Q1: 12.491, Q3: 224.409, IQR: 211.917
 Границы выбросов: [-305.385, 542.285]
 Количество выбросов: 145 (14.5%)
 Максимальное значение: 4128.529
 Минимальное значение: 0.004

CC50, mM:

Q1: 99.999, Q3: 891.771, IQR: 791.772
 Границы выбросов: [-1087.659, 2079.429]
 Количество выбросов: 39 (3.9%)
 Максимальное значение: 4538.976
 Минимальное значение: 0.701

SI:

Q1: 1.457, Q3: 16.525, IQR: 15.068
 Границы выбросов: [-21.144, 39.127]
 Количество выбросов: 124 (12.4%)
 Максимальное значение: 15620.600
 Минимальное значение: 0.011

Выполнен **анализ вариативности признаков** - диапазона изменений признака, что поможет исключить константы и почти стабильные признаки. Если некоторые компоненты или характеристики химических соединений практически не меняются, то это не поможет нам найти различия в целевых переменных (хотя возможна, например, ситуация, когда химическое вещество работает как катализатор)

Общее количество признаков: 211

Числовые признаки: 211

Признаки с нулевой вариативностью: 18

Признаки с низкой вариативностью: 0

Из 211 числовых признаков 18 имеют нулевую вариативность, что делает их непригодными для моделирования и требует удаления. Низкой вариативности не обнаружено.

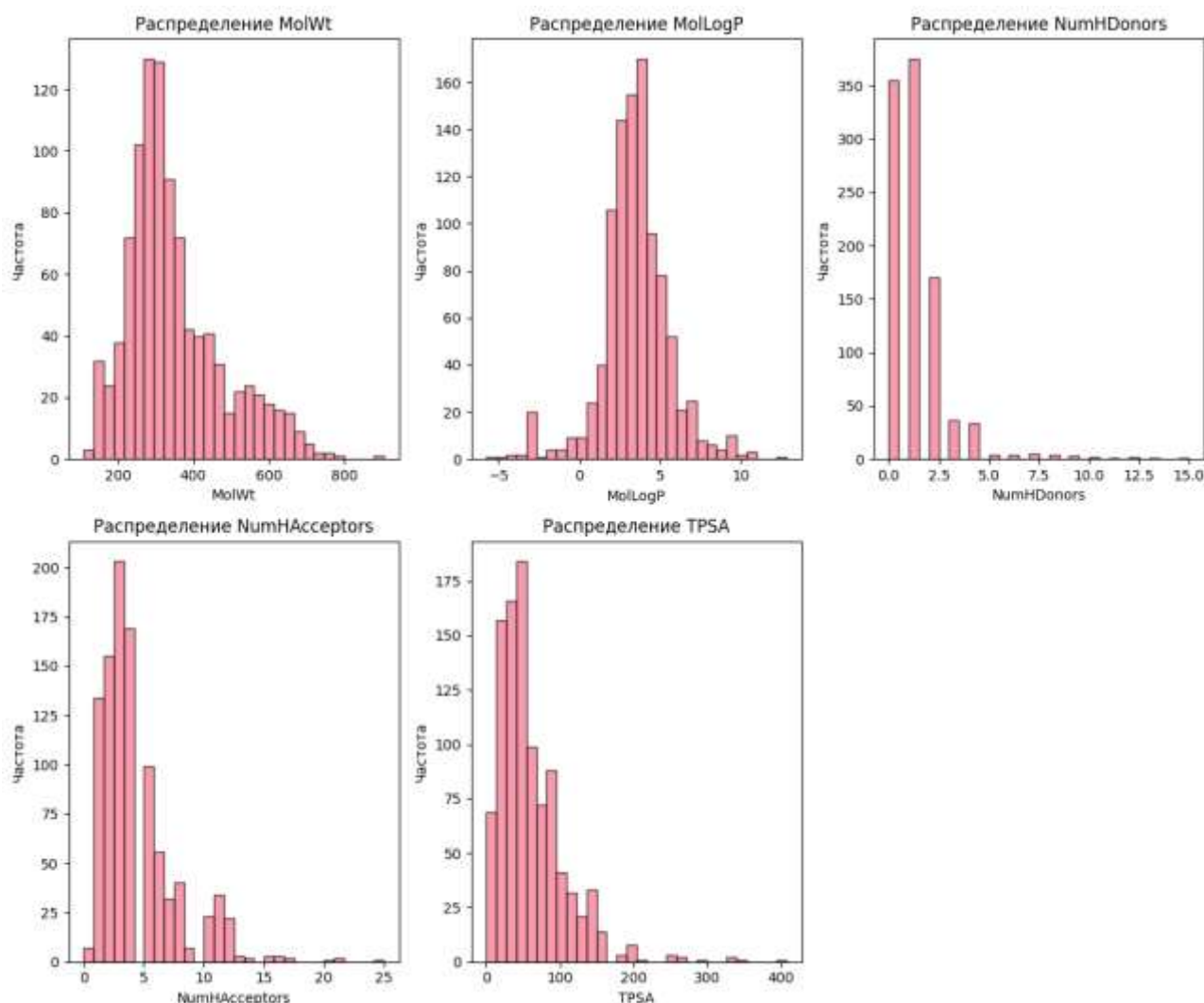
Проанализированы распределения некоторых признаков, например:

MolWt Молекулярная масса (Molecular Weight) - Слишком большие молекулы хуже проникают через клеточные мембраны и плохо абсорбируются

MolLogP - Логарифм распределения между октанолом и водой (гидрофобность) - растворимость и проницаемость вещества.

NumHDonors Количество доноров водородных связей и **NumHAcceptors** Количество акцепторов водородных связей влияют на биодоступность вещества

TPSA Полярная поверхность (Topological Polar Surface Area) - Связана с способностью к абсорбции и проникновению через биомембраны (включая гематоэнцефалический барьер)



Вывод: Из-за асимметричного распределения и наличия выбросов рекомендуется использовать модели, устойчивые к шуму и выбросам (например, градиентный бустинг, деревья решений) и применять масштабирование и/или преобразование признаков (логарифмирование)

Корреляционный анализ

Топ-10 признаков, наиболее коррелирующих с IC50, mM:

VSA_EState4	0.271743
Chi2n	0.252705
PEOE_VSA7	0.250772

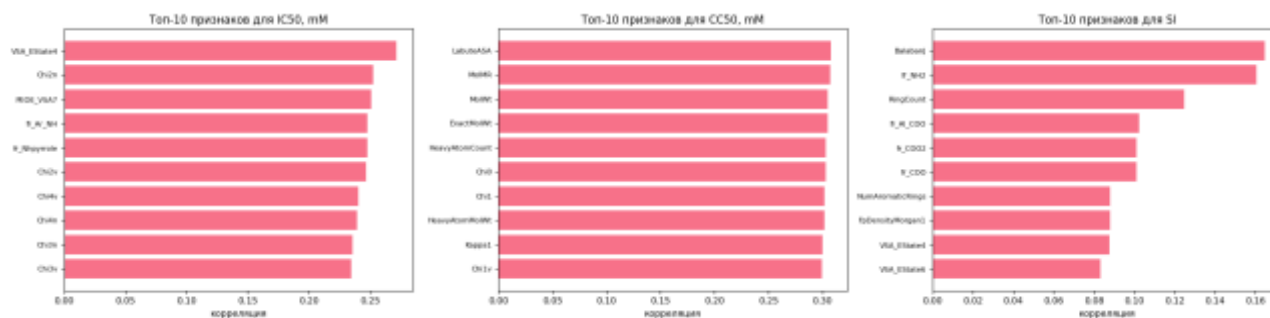
fr_Ar_NH	0.247728
fr_Nhpyrrole	0.247728
Chi2v	0.246602
Chi4v	0.240485
Chi4n	0.239209
Chi3n	0.235491
Chi3v	0.234816

Топ-10 признаков, наиболее коррелирующих с CC50, mM:

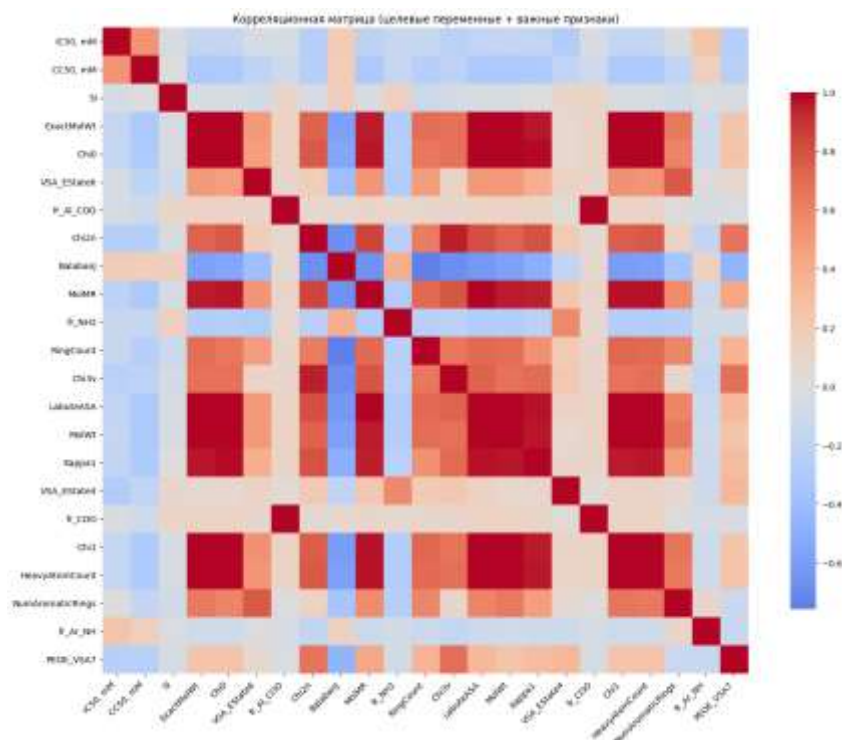
LabuteASA	0.307774
MolMR	0.307252
MolWt	0.304940
ExactMolWt	0.304883
HeavyAtomCount	0.303065
Chi0	0.302859
Chi1	0.302238
HeavyAtomMolWt	0.302112
Kappa1	0.300393
Chi1v	0.299828

Топ-10 признаков, наиболее коррелирующих с SI:

BalabanJ	0.164715
fr_NH2	0.160428
RingCount	0.124835
fr_Al_COO	0.102374
fr_COO2	0.101075
fr_COO	0.101075
NumAromaticRings	0.088006
FpDensityMorgan1	0.087894
VSA_EState4	0.087770
VSA_EState6	0.083298



- IC50, mM: Наибольшие корреляции у VSA_EState4 (0.271743), Chi2n (0.252705) и PEOE_VSA7 (0.250772).
- CC50, mM: Наиболее коррелирующие признаки включают LabuteASA (0.307774), MolMR (0.307252) и MolWt (0.304940).
- SI: Имеет значительно более низкие корреляции с признаками, среди которых BalabanJ (0.164715), fr_NH2 (0.160428) и RingCount (0.124835). Общий уровень корреляции признаков с SI ниже, чем с IC50 и CC50, что может указывать на более сложную нелинейную зависимость



По итогам EDA выполнен отбор признаков для каждой целевой переменной:

1. убраны признаки с нулевой и низкой вариативностью и низкой корреляцией с целевой переменной
2. удалены мультиколлинеарные признаки
3. отобраны признаки с помощью модели Random Forest.

Признаки, отобранные для всех целевых переменных

	IC50, mM	CC50, mM	SI
BCUT2D_LOGPLOW	1	1	1
BalabanJ	1	1	1
FpDensityMorgan1	1	1	1
MolLogP	1	1	1
VSA_EState4	1	1	1

На выходе получим датасет для каждой из 3-х целевых переменных с отобранными признаками

IC50, mM - selected_features_IC50, mM.csv
 CC50, mM - selected_features_CC50, mM.csv
 SI - selected_features_SI.csv

Вывод

- Исходный датасет очищен и сокращён до релевантных признаков.
- Были удалены шумовые и дублирующие признаки, учтена мультиколлинеарность.

- Подготовлены 3 оптимизированных датасета (selected_features_SI.csv, selected_features_IC50_mM.csv, selected_features_CC50_mM.csv) по каждому таргету для построения моделей

Решение задач регрессии IC50, CC50 и SI

В качестве исходных данных для решения задач **регрессии IC50, CC50 и SI** использовались датасеты, подготовленные на этапе анализа данных (EDA). Выполнено предварительно логарифмирование целевых переменных.

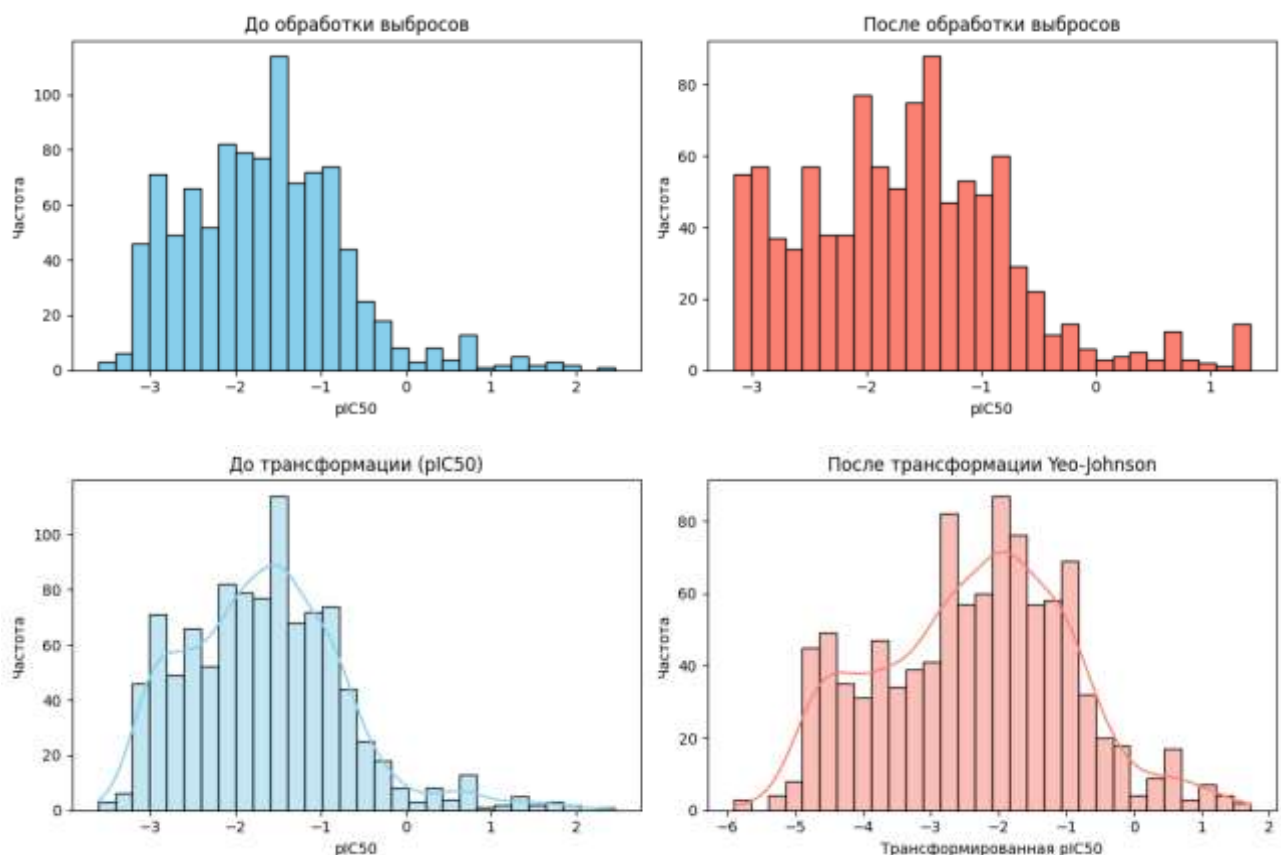
Регрессия для IC50

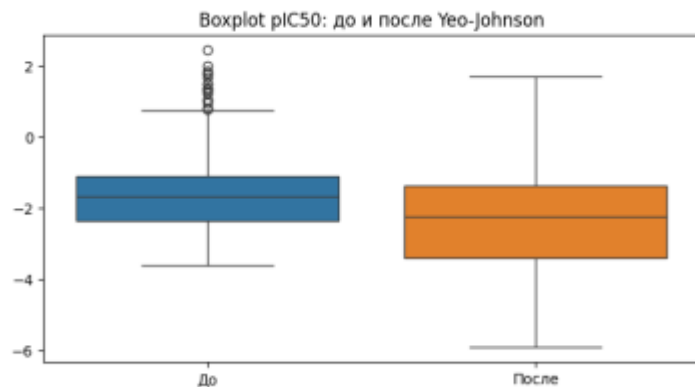
На основе логарифмированных данных проведено исследование влияния дополнительных трансформаций на распределение целевой переменной, включая устранение экстремальных значений через обрезку по перцентилям 1%-99% и нормализующее преобразование Yeo-Johnson.

Нормализация распределения целевой переменной критически важна для параметрических моделей (линейная регрессия, Ridge, Lasso), которые основаны на предположениях о нормальности остатков.

Эти методы чувствительны к выбросам и асимметрии данных, поэтому качественная предобработка напрямую влияет на их производительность.

В то же время, непараметрические алгоритмы машинного обучения, включая случайный лес, градиентный бустинг и нейронные сети, способны эффективно работать с исходными нетрансформированными данными, поскольку они не делают строгих предположений о распределении целевой переменной и могут самостоятельно адаптироваться к сложным нелинейным зависимостям в данных.

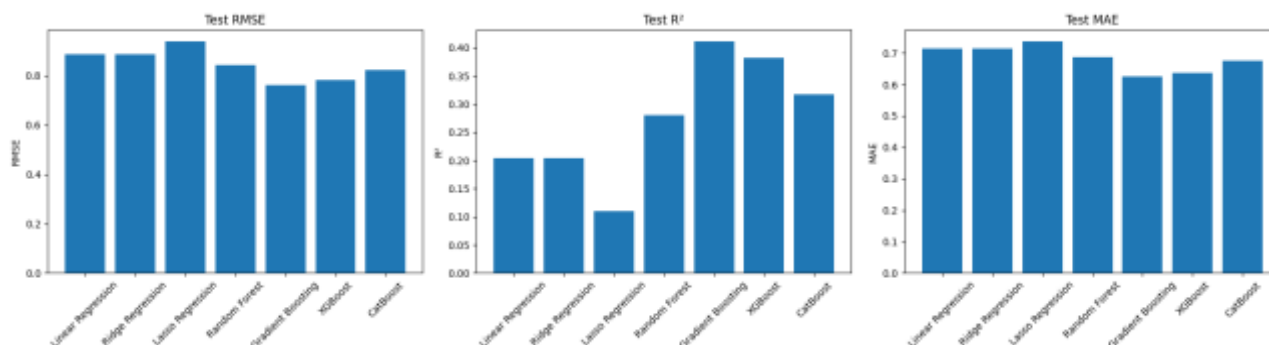




Результат работы моделей с первоначально заданными параметрами:

	Model	Train_RMSE	Test_RMSE	Train_MAE	Test_MAE	Train_R2	\
0	Linear Regression	0.7917	0.8869	0.6332	0.7147	0.2320	
1	Ridge Regression	0.7917	0.8869	0.6332	0.7147	0.2320	
2	Lasso Regression	0.8509	0.9375	0.6739	0.7357	0.1129	
3	Random Forest	0.7047	0.8433	0.5744	0.6873	0.3915	
4	Gradient Boosting	0.6235	0.7631	0.5142	0.6258	0.5236	
5	XGBoost	0.6325	0.7811	0.5198	0.6375	0.5097	
6	CatBoost	0.7153	0.8217	0.5837	0.6755	0.3731	

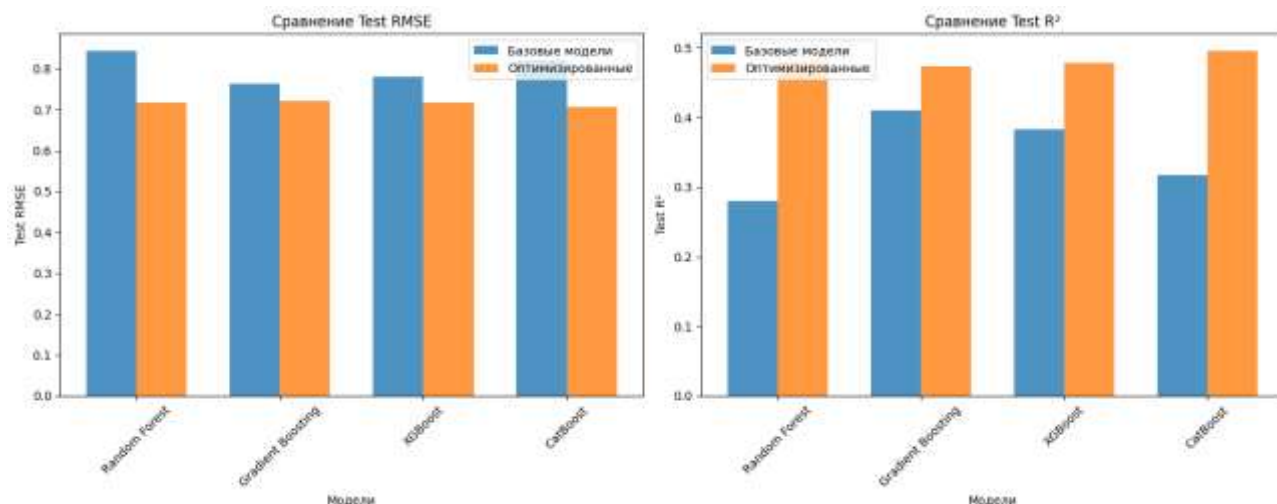
	Test_R2	Test_Acc(±0.5)	CV_RMSE
0	0.2039	0.410	0.8231
1	0.2040	0.405	0.8228
2	0.1105	0.400	0.8589
3	0.2802	0.425	0.7443
4	0.4107	0.440	0.7157
5	0.3826	0.455	0.7139
6	0.3167	0.420	0.7517



В результате анализа работы различных моделей регрессии с заданными параметрами, можно выделить Gradient Boosting и XGBoost как наиболее эффективные. Эти модели показали наименьшие значения метрик Test RMSE и Test MAE, а также наивысшие значения Test R2, что указывает на их лучшую способность предсказывать целевую переменную и объяснять дисперсию данных.

В частности, Gradient Boosting демонстрирует Test RMSE 0.6235, Test MAE 0.6258 и Test R2 0.5231, что является лучшим результатом среди всех моделей. XGBoost также показывает очень хорошие результаты, уступая Gradient Boosting лишь незначительно. Остальные модели, такие как Linear Regression, Ridge Regression, Lasso Regression и Random Forest, показали худшие результаты по сравнению с бустинговыми алгоритмами. CatBoost также показал себя хуже, чем Gradient Boosting и XGBoost.

Подбор гиперпараметров с использованием Optuna



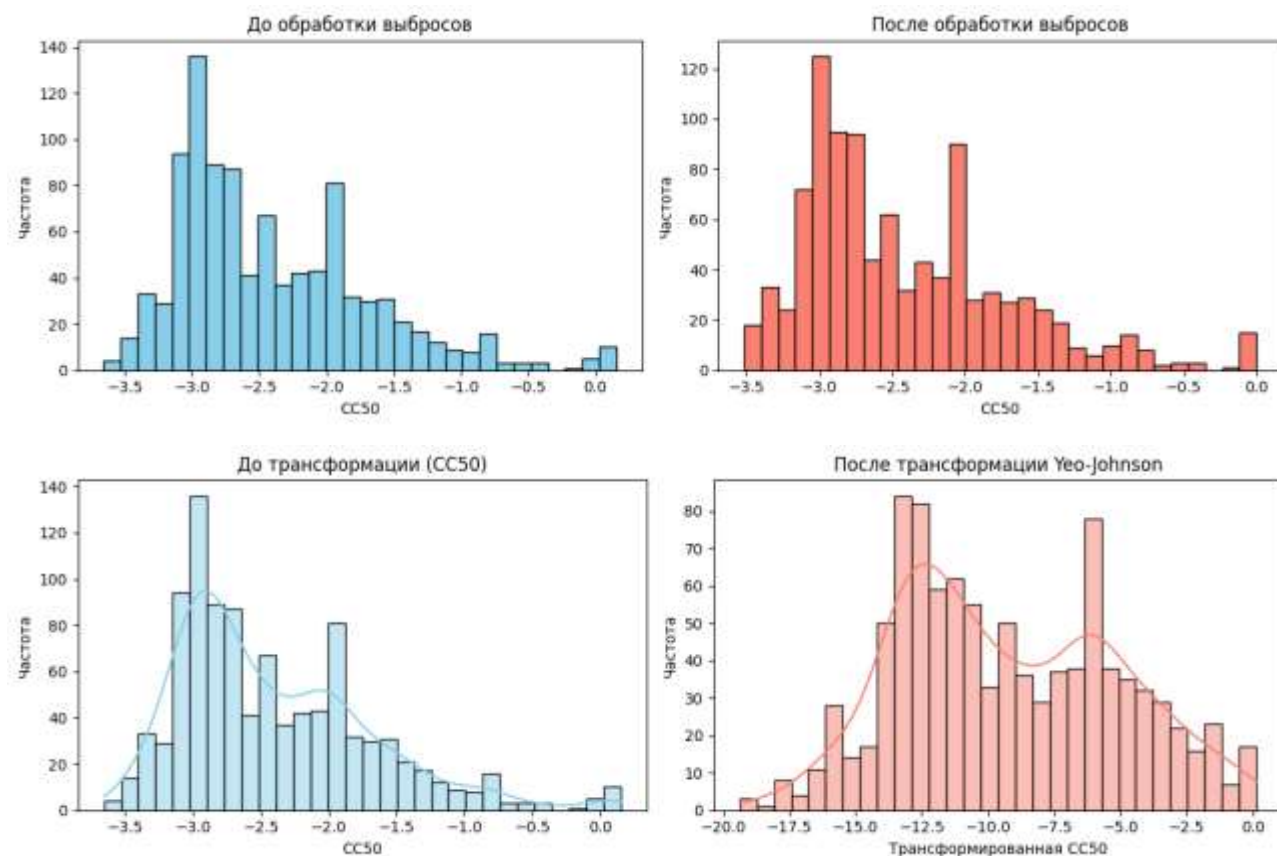
Результат ясно демонстрируют критическую важность оптимизации гиперпараметров. **Gradient Boosting** и **XGBoost** улучшили производительность после настройки. CatBoost, который ранее отставал, показывает значительный скачок в производительности после оптимизации, что приближает его к лучшим моделям бустинга.

Т.о., для данной задачи **XGBoost** и **Gradient Boosting** (после оптимизации) являются лучшими моделями. Они показывают наименьший **Test RMSE** и наибольший **Test R²**.

CatBoost после оптимизации также может быть рассмотрен.

Регрессия для CC50

Исследование влияния дополнительных трансформаций на распределение целевой переменной, включая устранение экстремальных значений через обрезку по процентиллям 1%-99% и нормализующее преобразование Yeo-Johnson.



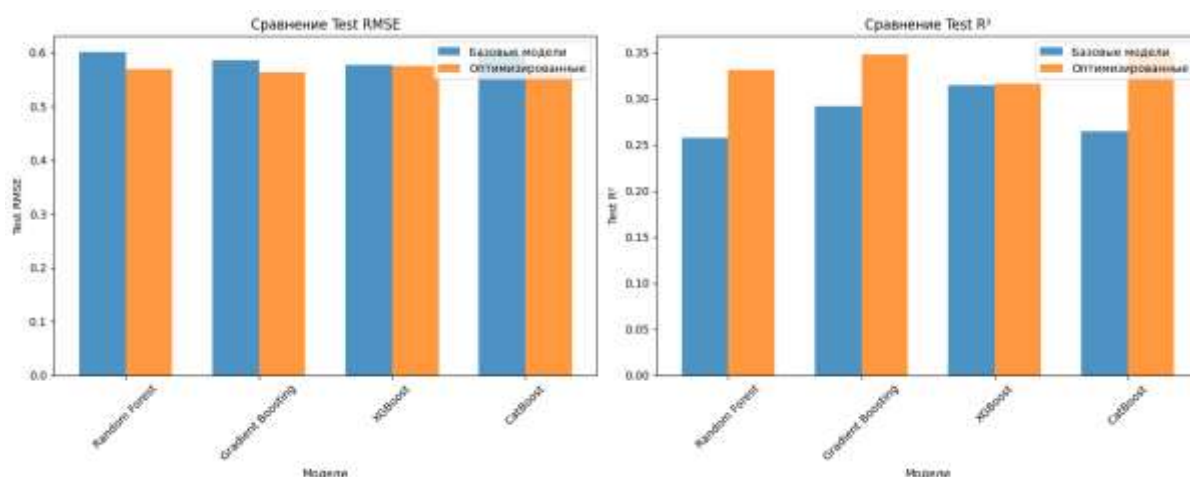
Результат работы базовых моделей

Model Train_RMSE Test_RMSE Train_MAE Test_MAE Train_R2 \

0	Linear Regression	0.5956	0.6044	0.4516	0.4803	0.3037
1	Ridge Regression	0.5956	0.6043	0.4517	0.4807	0.3036
2	Lasso Regression	0.6742	0.6704	0.5361	0.5542	0.1078
3	Random Forest	0.5648	0.6006	0.4429	0.4852	0.3739
4	Gradient Boosting	0.4999	0.5865	0.3864	0.4600	0.5094
5	XGBoost	0.5014	0.5771	0.3834	0.4481	0.5065
6	CatBoost	0.5732	0.5979	0.4509	0.4851	0.3550

	Test_R2	Test_Acc(± 0.5)	CV_RMSE
0	0.2482	0.605	0.6156
1	0.2484	0.615	0.6153
2	0.0749	0.515	0.6769
3	0.2577	0.635	0.6000
4	0.2920	0.640	0.5630
5	0.3147	0.655	0.5662
6	0.2643	0.620	0.5984

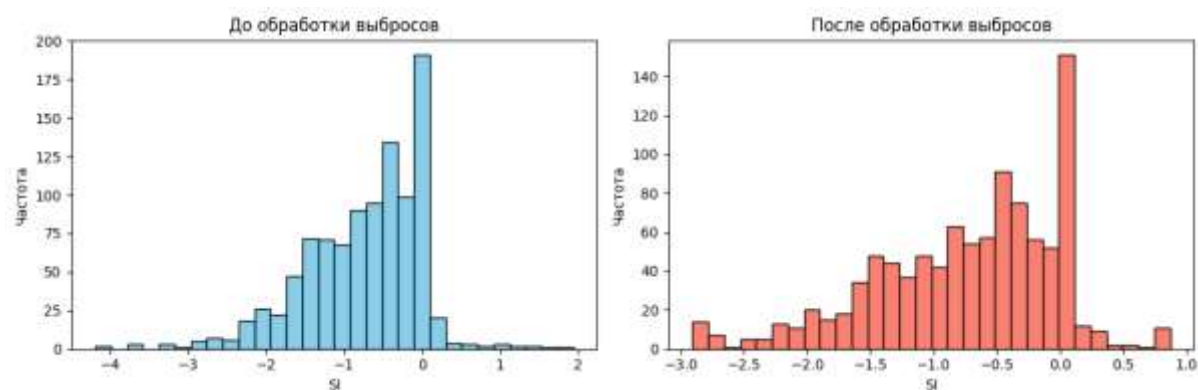
Подбор гиперпараметров с использованием Optuna

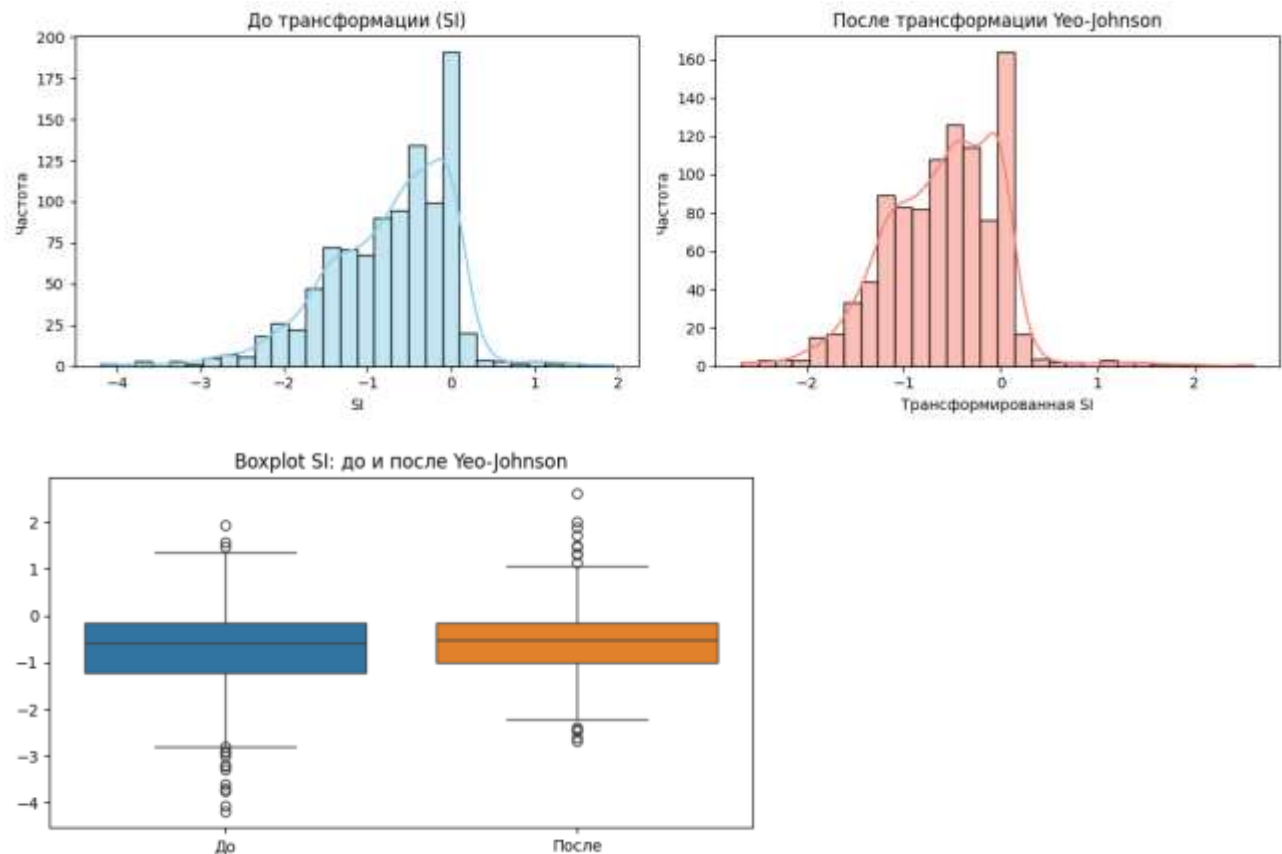


Аналогично регрессии для IC50 данные результаты показывают, что оптимизация гиперпараметров улучшила результаты моделей **Gradient Boosting** и **XGBoost**. **CatBoost**, после оптимизации гиперпараметров значительно улучшил результат.

Регрессия для IS

Исследование распределение целевой переменной





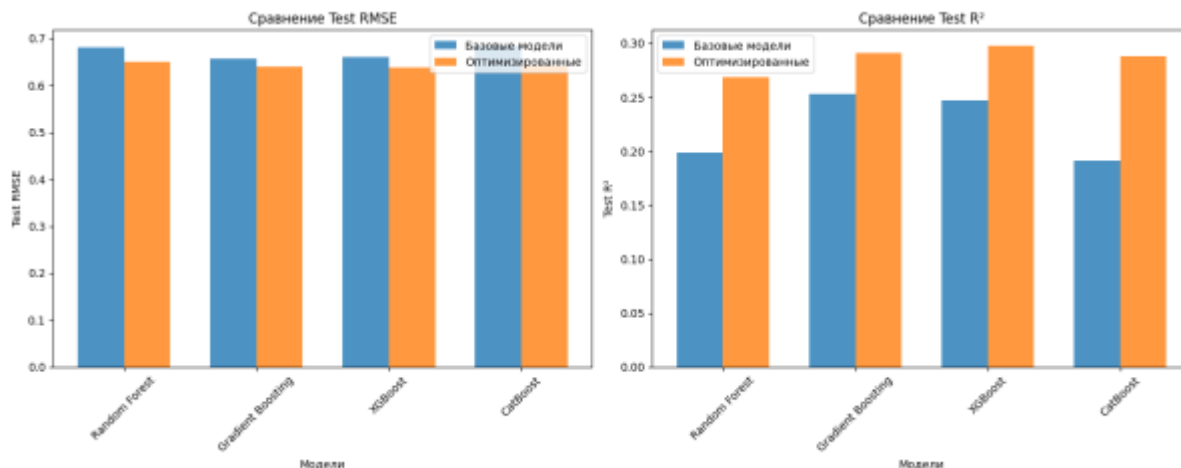
Результат работы моделей с первоначально заданными параметрами:

	Model	Train_RMSE	Test_RMSE	Train_MAE	Test_MAE	Train_R2	\
0	Linear Regression	0.6469	0.7057	0.5131	0.5638	0.1729	
1	Ridge Regression	0.6469	0.7057	0.5131	0.5639	0.1729	
2	Lasso Regression	0.6945	0.7461	0.5608	0.6112	0.0468	
3	Random Forest	0.6038	0.6805	0.4791	0.5429	0.2795	
4	Gradient Boosting	0.5497	0.6569	0.4324	0.5217	0.4027	
5	XGBoost	0.5560	0.6594	0.4357	0.5256	0.3890	
6	CatBoost	0.6102	0.6835	0.4853	0.5500	0.2642	

	Test_R2	Test_Acc(±0.5)	CV_RMSE
0	0.1381	0.540	0.6673
1	0.1380	0.540	0.6670
2	0.0366	0.445	0.6964
3	0.1984	0.515	0.6391
4	0.2531	0.555	0.6186
5	0.2475	0.565	0.6211
6	0.1914	0.500	0.6408

Подбор гиперпараметров с использованием Optuna

Способность распознавать (предсказывать) тестовые данные улучшилась у всех моделей, для которых был проведен подбор гиперпараметров. Уменьшилась среднеквадратичная ошибка на тестовых данных **Test RMSE** и увеличилось значение **Test R²** - что доля дисперсии целевой переменной, объясняемая моделью на тестовых данных, стала больше.



Выводы:

1. Непараметрические алгоритмы машинного обучения показали лучшие результаты, чем параметрические модели. Этот результат ожидаем, поскольку данные физико-химических исследований, как правило, имеют нелинейные зависимости и распределения, далекие от нормального.
2. Подбор гиперпараметров является критическим шагом для построения эффективной модели. Каким бы тщательным ни был ручной подбор первичных параметров, после автоматического подбора гиперпараметров модели стали более точными в своих предсказаниях на новых данных (ниже RMSE) и лучше объясняют их вариативность (выше R^2).
3. При построении регрессии для всех целевых переменных в ряде случаев наблюдалось критическое снижение метрики R^2 на тестовых данных. Это говорит о наличии переобучения модели, а также об избыточной архитектуре (например, слишком большое количество деревьев) для данной задачи, когда модель не учится, а запоминает примеры. Целесообразно провести дальнейшую работу по оптимизации архитектуры применяемых моделей.
4. В данной работе умышленно использовалось небольшое количество параметров (24) по сравнению с исходным набором (211) для экономии времени и вычислительных ресурсов. Для практического применения необходимо расширить этот перечень и использовать приемы создания новых признаков (Feature Engineering) на основе предоставленных данных.

Классификация:

превышает ли значение IC50, CC50 и IS

медианное значение выборки

Алгоритм решения одинаков для каждой из 3-х задач:

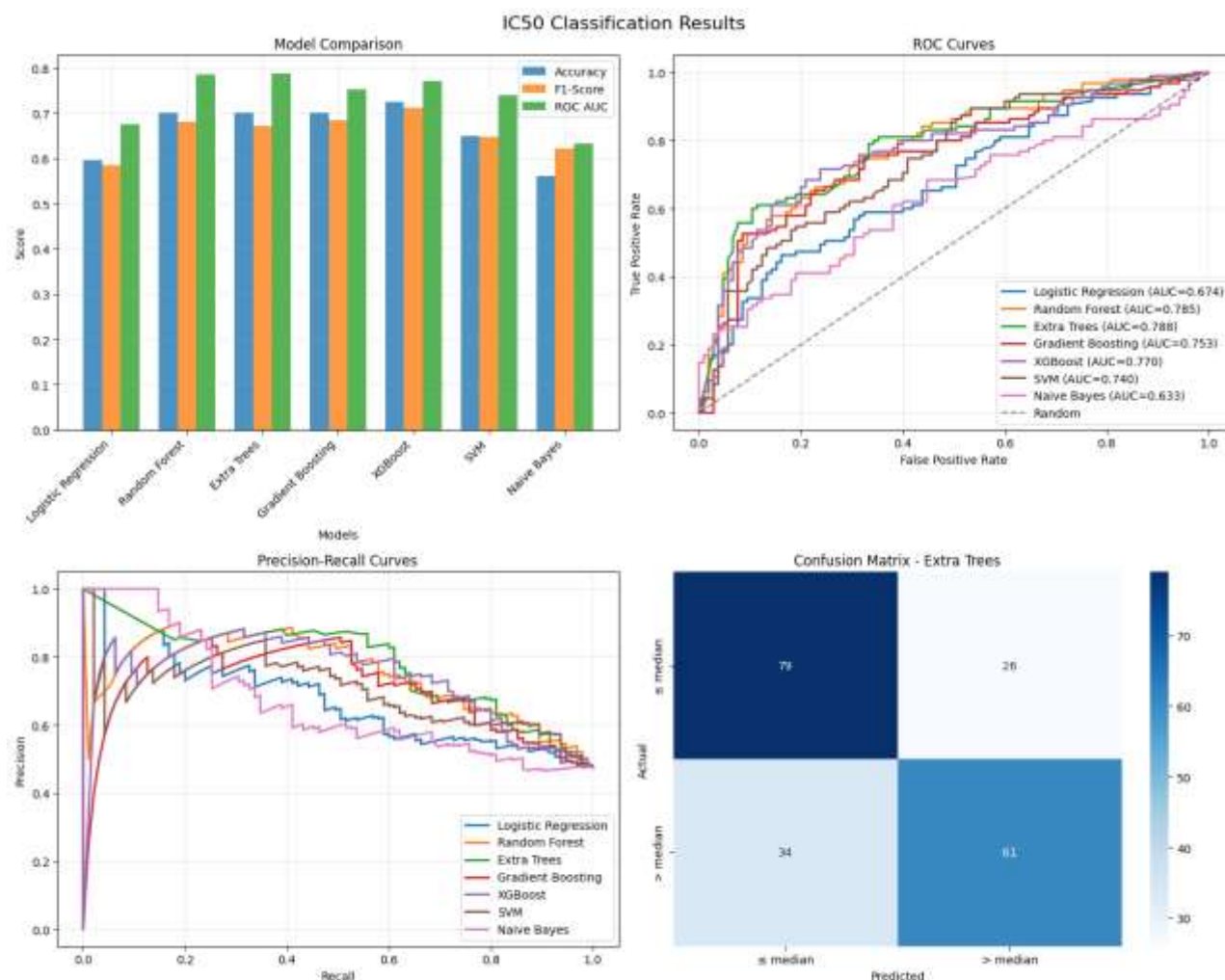
- Используем предварительно подготовленный датасет (как в задачах для регрессии)
- Делим на train-test с непрерывной целевой переменной
- Считаем медиану **по train**.
- **Создаем бинарные метки** для train и test на основе этой медианы

- Обучаем модели и тестируем
- Сравниваем модели по метрикам

В работе применялись модели:

Модель	Краткая характеристика	Обоснование применимости к задаче
Logistic Regression	Линейная модель для бинарной классификации, использующая логистическую функцию	Хорошо подходит для понимания влияния отдельных дескрипторов на превышение медианы. Интерпретируемые коэффициенты показывают, какие химические свойства увеличивают/уменьшают вероятность превышения
Random Forest	Ансамбль деревьев решений с бэггингом и случайным выбором признаков	Отлично подходит для химических данных с множественными нелинейными взаимодействиями между дескрипторами. Может выявлять сложные структурно-активные взаимосвязи
Extra Trees	Ансамбль экстремально рандомизированных деревьев	Подходит для обнаружения скрытых паттернов в молекулярных дескрипторах благодаря дополнительной рандомизации. Может лучше генерализовать на новые химические структуры
Gradient Boosting	Последовательное обучение слабых моделей с исправлением ошибок	Эффективен для выявления сложных нелинейных зависимостей "структура-активность". Последовательное обучение позволяет точно моделировать биологическую активность
XGBoost	Оптимизированная реализация градиентного бустинга	Один из лучших алгоритмов для табличных данных. Регуляризация помогает избежать переобучения на химических данных. Отлично работает с молекулярными дескрипторами
SVM	Метод опорных векторов с RBF ядром для нелинейной классификации	Эффективен для высокоразмерных химических данных. RBF ядро может выявлять сложные нелинейные границы между активными и неактивными соединениями
Naive Bayes	Вероятностный классификатор на основе теоремы Байеса с предположением независимости признаков	Подходит как базовая модель для сравнения. Может работать с малыми выборками, но предположение независимости дескрипторов часто нарушается в химических данных

Классификация: превышает ли значение IC50 медианное значение выборки



Сводная таблица предсказаний:

Model	True_Class_0 (≤ медианы)	True_Class_1 (> медианы)	Pred_Class_0 (≤ медианы)	Pred_Class_1 (> медианы)	Correct	Total	Accuracy
Logistic Regression	105	95	100	100	119	200	0.595
Random Forest	105	95	107	93	140	200	0.700
Extra Trees	105	95	113	87	140	200	0.700
Gradient Boosting	105	95	105	95	140	200	0.700
XGBoost	105	95	104	96	145	200	0.725
SVM	105	95	97	103	130	200	0.650
Naive Bayes	105	95	63	137	112	200	0.560

ЛУЧШАЯ МОДЕЛЬ: Extra Trees

- ROC AUC: 0.7877
- Accuracy: 0.7000
- F1-Score: 0.6703

79 - True Negatives (правильно предсказан класс "≤ median")

26 - False Positives (ошибочно предсказан как "> median")

34 - False Negatives (ошибочно предсказан как "≤ median")

61 - True Positives (правильно предсказан класс "> median")

Accuracy = $(79 + 61) / (79 + 26 + 34 + 61) = 140/200 = 70\%$

- Класс "≤ median": $79/(79+26) = 75.2\%$ - хорошо
- Класс "> median": $61/(34+61) = 64.2\%$ - средне

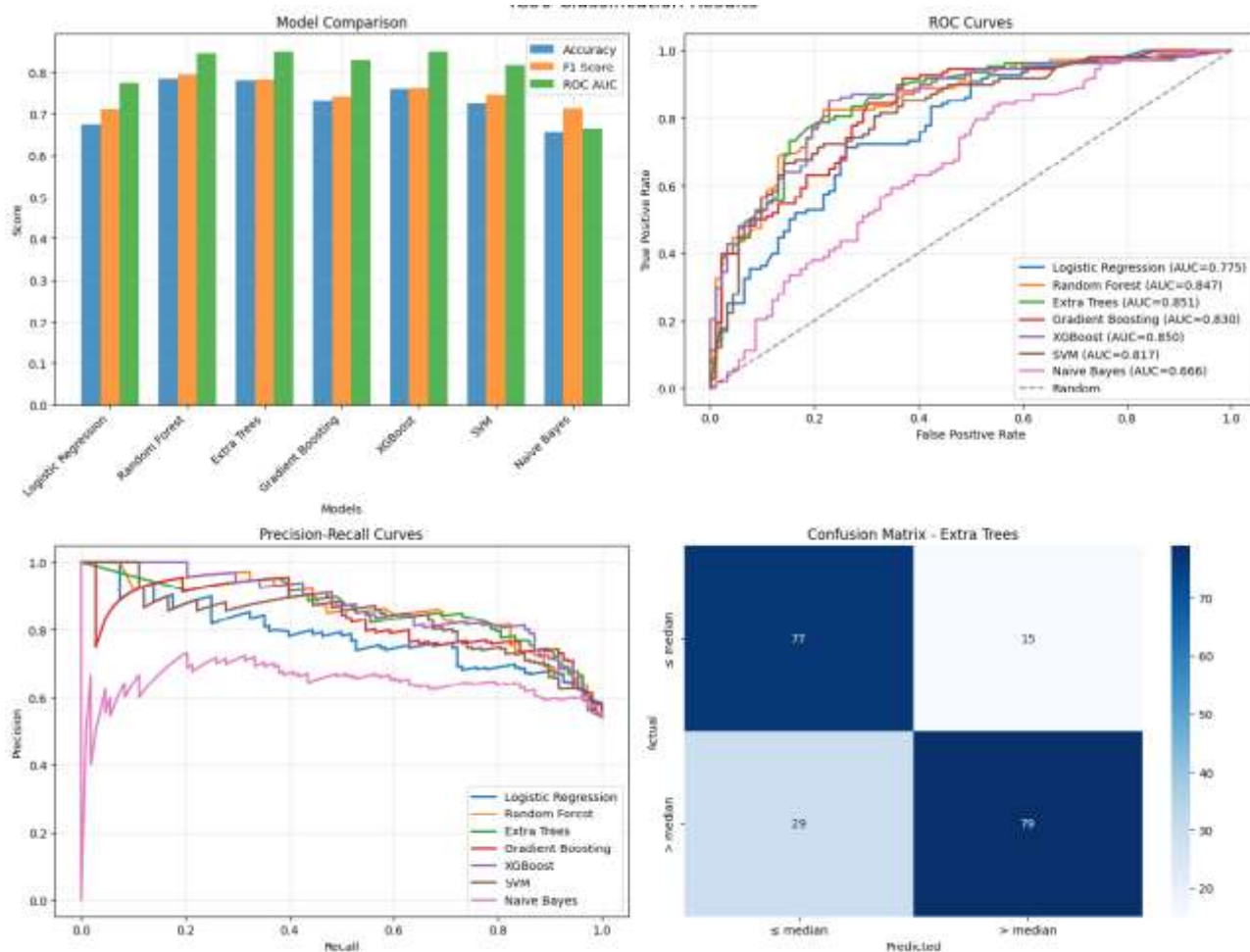
Полнота (Recall):

- Класс "≤ median": $79/(79+34) = 69.9\%$
- Класс "> median": $61/(26+61) = 70.1\%$

Модель неплохо разделила классы (70% точность), правильных предсказаний больше, чем ошибок. Но модель допустила 26 ложных срабатываний - соединения с $IC_{50} \leq$ медианы модель считает активными, и 34 пропуска - активные соединения ($>$ медианы) модель считает неактивными.

Т.е. мы ложно отбираем 26 соединений и упускаем 34 потенциально активных.

Классификация: превышает ли значение CC_{50} медианное значение выборки



Сводная таблица предсказаний:

Model	True_Class_0 (≤ медианы)	True_Class_1 (> медианы)	Pred_Class_0 (≤ медианы)	Pred_Class_1 (> медианы)	Correct	Total	Accuracy
Logistic Regression	92	108	83	117	135	200	0.675
Random Forest	92	108	99	101	157	200	0.785
Extra Trees	92	108	106	94	156	200	0.780
Gradient Boosting	92	108	98	102	146	200	0.730
XGBoost	92	108	106	94	152	200	0.760
SVM	92	108	91	109	145	200	0.725
Naive Bayes	92	108	67	133	131	200	0.655

ЛУЧШАЯ МОДЕЛЬ: Extra Trees

- ROC AUC: 0.8506
- Accuracy: 0.7800
- F1-Score: 0.7822

77 - True Negatives (правильно предсказан класс " \leq median")

15 - False Positives (ошибочно предсказан как " $>$ median")

29 - False Negatives (ошибочно предсказан как " \leq median")

79 - True Positives (правильно предсказан класс " $>$ median")

Accuracy = $(77 + 79) / (77 + 15 + 29 + 79) = 156/200 = 78\%$

- Класс " \leq median": $77/(77+29) = 72.6\%$ - хорошо

- Класс "> median": $79/(15+79) = 84.0\%$ - отлично

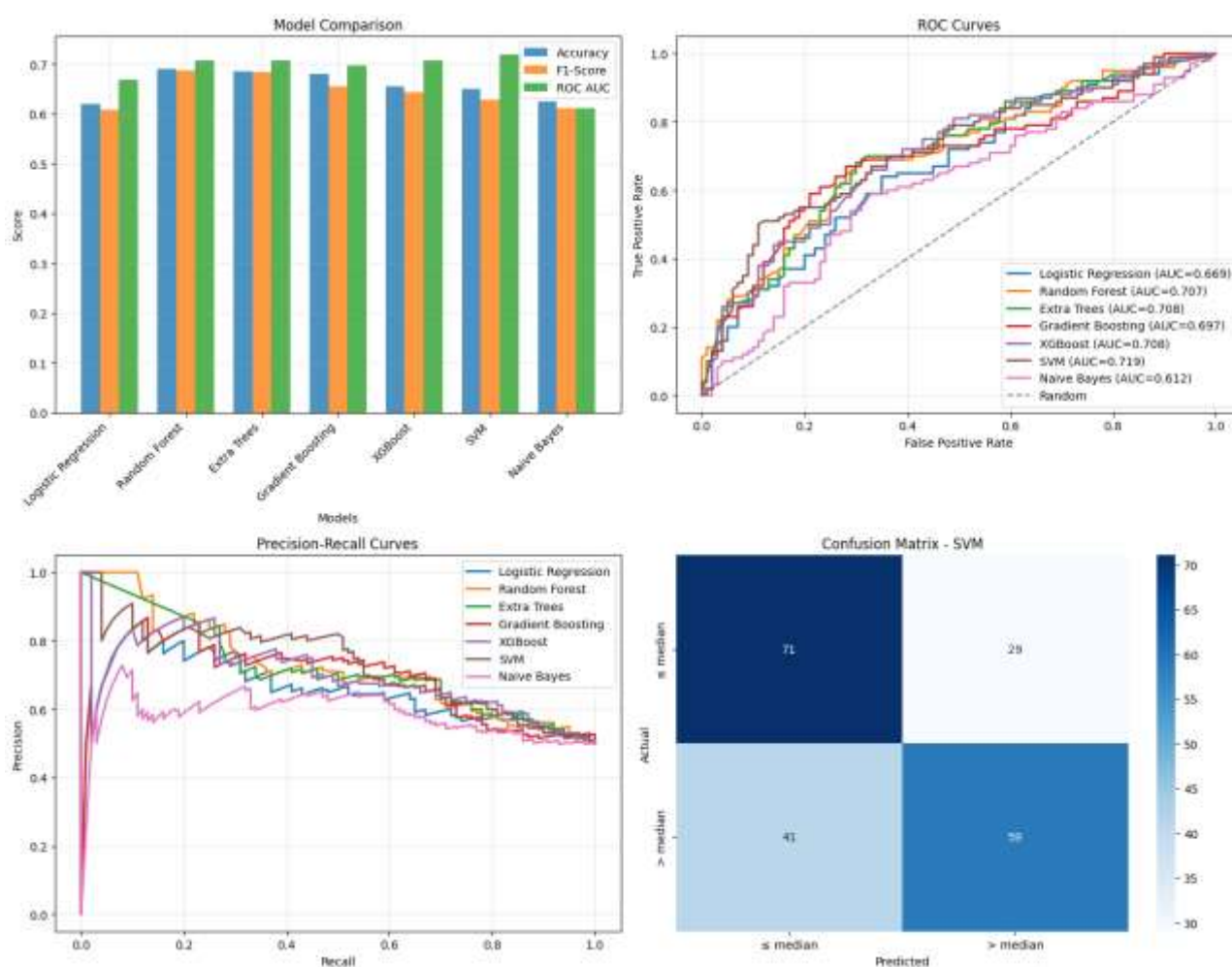
Полнота (Recall):

- Класс " \leq median": $77/(77+15) = 83.7\%$
- Класс "> median": $79/(29+79) = 73.1\%$

Модель неплохо разделила классы (78% точность), правильных предсказаний больше, чем ошибок. Но модель допустила 15 ложных срабатываний - соединения с $IC_{50} \leq$ медианы модель считает активными, и 29 пропусков - активные соединения ($>$ медианы) модель считает неактивными.

Т.е. мы ложно отбираем 15 соединений и упускаем 29 потенциально активных

Классификация: превышает ли значение SI медианное значение выборки



Сводная таблица предсказаний:

Model	True_Class_0 (\leq медианы)	True_Class_1 ($>$ медианы)	Pred_Class_0 (\leq медианы)	Pred_Class_1 ($>$ медианы)	Correct	Total	Accuracy
Logistic Regression	100	100	106	94	124	200	0.620
Random Forest	100	100	102	98	138	200	0.690
Extra Trees	100	100	101	99	137	200	0.685
Gradient Boosting	100	100	114	86	136	200	0.680
XGBoost	100	100	107	93	131	200	0.655
SVM	100	100	112	88	130	200	0.650
Naive Bayes	100	100	107	93	125	200	0.625

ЛУЧШАЯ МОДЕЛЬ: SVM

- ROC AUC: 0.7194
- Accuracy: 0.6500
- F1-Score: 0.6277

71 - True Negatives (правильно предсказан класс " \leq median")

29 - False Positives (ошибочно предсказан как "> median")

41 - False Negatives (ошибочно предсказан как " \leq median")

59 - True Positives (правильно предсказан класс "> median")

Accuracy = $(71 + 59) / (71 + 29 + 41 + 59) = 130/200 = 65\%$

- Класс " \leq median": $71/(71+41) = 63.4\%$ - средне
- Класс "> median": $59/(29+59) = 67.0\%$ - средне

Полнота (Recall):

- Класс " \leq median": $71/(71+29) = 71.0\%$ - хорошо
- Класс "> median": $59/(41+59) = 59.0\%$ - средне

Точность разделения на классы 65% точность

Допущено 29 ложных срабатываний - соединения с $IC_{50} \leq$ медианы модель считает активными, и 41 пропуск - активные соединения (> медианы) модель считает неактивными.

Т.е. мы ложно отбираем 29 соединений и упускаем 41 потенциально активных

Выводы:

Наилучшие результаты получены с помощью моделей **SVM (с RBF-ядром)** и **Extra Trees**.

SVM эффективно работает в высокоразмерном пространстве химических дескрипторов, создавая нелинейные границы решений.

Extra Trees благодаря дополнительной рандомизации лучше генерализует на новые химические структуры и устойчив к переобучению на сложных молекулярных данных.

Классификация: превышает ли значение SI > 8

SI (Индекс селективности)

Практическое значение SI = 8 - выше этого порога значительно увеличиваются шансы на клинический успех лекарства

SI < 2: Общая токсичность (неприемлемо)

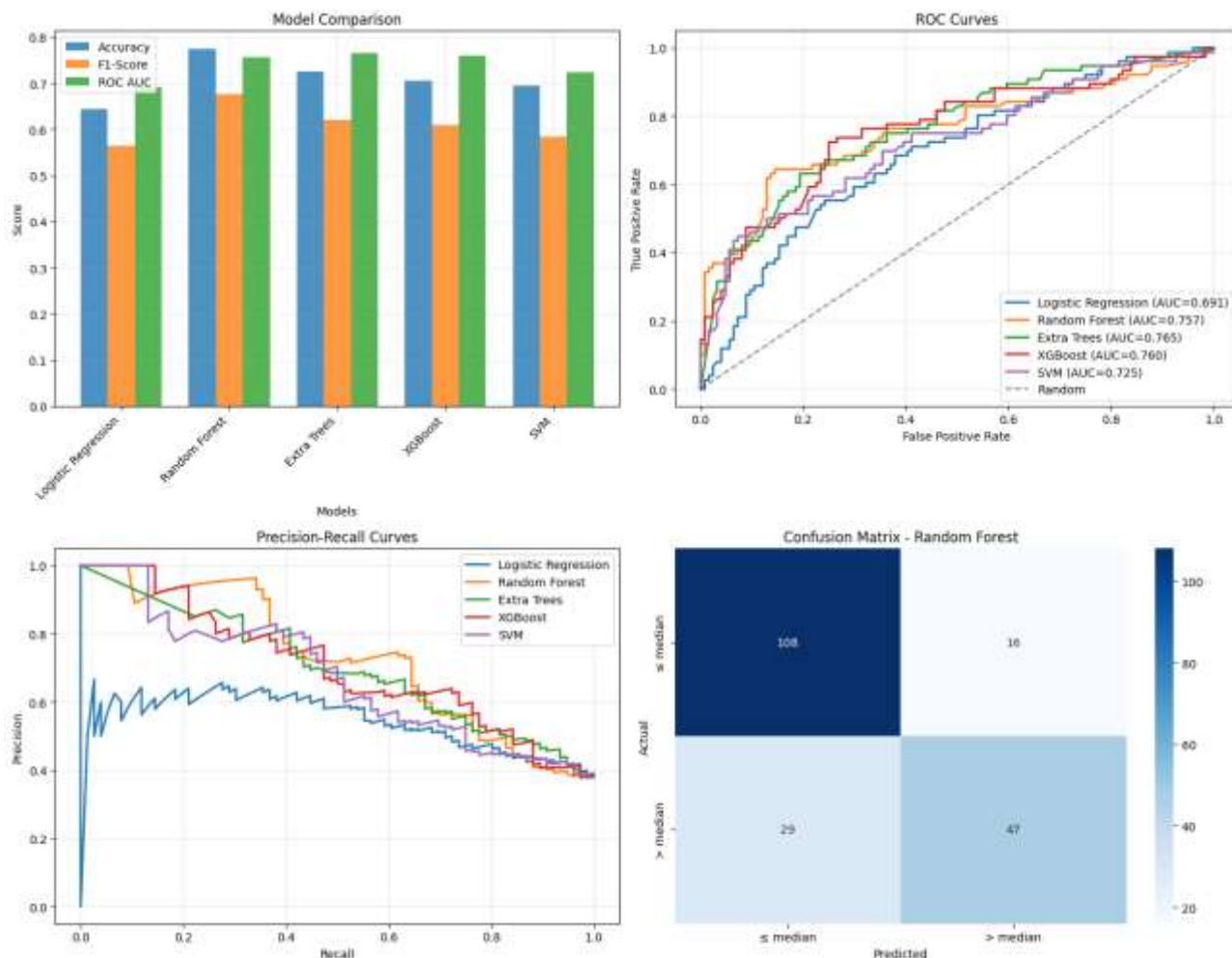
SI 2-5: Низкая селективность

SI 5-10: Умеренная селективность

SI > 8-10: Хорошая селективность

В датасете умеренный дисбаланс классов 64.9% / 35.1%, класс "высокоселективных" соединений (SI > 8), используем class_weight='balanced'

Сводная таблица предсказаний:								
Model	True_Class_0 (≤ 8)	True_Class_1 (> 8)	Pred_Class_0 (≤ 8)	Pred_Class_1 (> 8)	Correct	Total	Accuracy	
Logistic Regression	124	76	113	87	129	200	0.645	
Random Forest	124	76	137	63	155	200	0.775	
Extra Trees	124	76	131	69	145	200	0.725	
XGBoost	124	76	125	75	141	200	0.705	
SVM	124	76	129	71	139	200	0.695	



Для оценки и оптимизации моделей использовалась **стратифицированная 5-кратная кросс-валидация** с перемешиванием данных (StratifiedKFold, random_state=42), которая обеспечивает сохранение пропорций классов в каждой складке и повышает надежность оценки качества модели. **Подбор гиперпараметров** выполнялся методом **Grid Search** с оптимизацией по метрике F1-score.

```
Logistic Regression...
  Время: 49с | F1: 0.564 → 0.564 (+0.0%)

Random Forest...
  Время: 49с | F1: 0.676 → 0.640 (-5.4%)

Extra Trees...
  Время: 49с | F1: 0.621 → 0.627 (+1.0%)

XGBoost...
  Время: 49с | F1: 0.609 → 0.597 (-2.0%)

SVM...
  Время: 49с | F1: 0.585 → 0.615 (+5.2%)
```

При подборе гиперпараметров результат для большинства моделей ухудшился.

Это связано с переобучением при подборе гиперпараметров на небольшой выборке (988 наблюдений) и агрессивной селекцией признаков (сокращение с 211 до 24 дескрипторов), что могло привести к потере важной информации о молекулярных свойствах.

Для улучшения результата рекомендуется изменить логику отбора признаков, расширить их количество и использовать прием создания новых признаков (Feature Engineering) на основе предоставленных.