Тараканов Александр

Курсовая Работа: Классическое Машинное обучение

Введение

Целью данного проекта является построение предсказательных моделей на основе классических методов машинного обучения (Classical ML) для анализа биологических свойств химических соединений. Задачи охватывают как регрессионные, так и классификационные сценарии, направленные на оценку эффективности, токсичности и селективности веществ, потенциально применимых в фармацевтических исследованиях.

В основе проекта лежит обработка и моделирование экспериментальных данных, содержащих значения ключевых биологических метрик:

- **IC50** — концентрация вещества, при которой достигается 50% ингибирование активности биологической цели (эффективность);
- **CC50** — концентрация, при которой вещество проявляет токсичность в 50% случаев (токсичность);
- **SI** (Selectivity Index) — отношение CC50 к IC50, отражающее баланс между эффективностью и безопасностью.

Работа включает полный цикл машинного обучения:

1. **Разведочный анализ данных (EDA)**: визуализация распределений, удаление выбросов, анализ корреляций и пропусков.
2. **Feature Engineering**: генерация новых признаков, логарифмирование, масштабирование.
3. **Решение задач классификации и регрессии**: для каждого из показателей обучались отдельные модели.
4. **Оценка моделей**: по метрикам Accuracy, F1, ROC AUC (для классификации) и MAE, RMSE, R^2 (для регрессии).
5. **Sanity-check**: сквозная валидация моделей на отложенной выборке, визуализация результатов и финальный анализ.

Проект выполнен в соответствии с требованиями курсовой работы по модулю *Классическое машинное обучение* и полностью воспроизводим. Все этапы оформлены в отдельных скриптах, артефакты моделей сохранены, результаты визуализированы. Финальные модели готовы к практическому применению, например, в задачах

скрининга лекарственных соединений или в ранней оценке профиля безопасности новых молекул.

Подготовка, структура проекта и ход работы

Перед тем как приступить к построению моделей, я прошёл длительный и методичный путь предварительной подготовки. Целью было не просто реализовать код, а глубоко понять задачу, структурировать подход и выстроить надёжный воспроизводимый пайплайн, соответствующий всем требованиям курса и практики классического машинного обучения.

Исследование задачи и постановка целей

На первых этапах было важно разобраться:

- Что именно измеряют переменные IC50, CC50 и SI;
- Какие преобразования уместны с учётом природы данных;
- Как формировать целевые переменные для задач классификации и регрессии;
- Какие признаки могут оказаться избыточными или, наоборот, ключевыми.

Особое внимание было уделено валидации значений SI как отношения CC50 / IC50 — вручную проверены десятки значений, расхождений не обнаружено, что подтвердило корректность входных данных.

Также был проведён анализ требований к проекту: соблюдение структуры, разделение train/test, отсутствие утечек данных, наличие визуализаций, сохранение артефактов и финальная проверка через скрипт `sanity_check.py`.

Структура проекта

В процессе работы проект был организован в виде модульной системы, где каждая задача реализована в отдельном Python-скрипте, а общие артефакты сохраняются в подпапки:

```
CML_Final/
├── data/                  # Подготовленные данные
│   └── data_prepared.csv     # Очищенный датасет с инженерными признаками
├── models/                # Обученные модели и препроцессинг (joblib)
├── plots/                 # ROC-кривые, scatter-графики и др.
├── eda.py                 # Разведочный анализ данных
├── clf_ic50_median.py       # Классификация IC50 > медианы
├── clf_cc50_median.py       # Классификация CC50 > медианы
├── clf_si_median.py         # Классификация SI > медианы
├── clf_si_gt8.py            # Классификация SI > 8
├── reg_ic50.py              # Регрессия log(IC50)
├── reg_cc50.py              # Регрессия log(CC50)
├── reg_si.py                # Регрессия log(SI)
├── sanity_check.py          # Финальная проверка всех моделей
├── main.py                # Единая точка запуска пайплайна
└── report.docx            # Итоговый аналитический отчёт
```

Все модели, скейлеры и списки признаков сохраняются отдельно и повторно используются при финальной валидации. Код написан с учётом повторного запуска и полной воспроизводимости.

Ход работы и этапы итераций

Работа шла поэтапно. Вначале фокус был на выполнении baseline-решения с простыми моделями и кросс-валидацией. Однако результаты показались недостаточными:

- Возникли подозрения на утечки данных (одни и те же данные использовались и для обучения, и для оценки).
- Метрики были слишком высоки — решено было перестроить пайплайн с жёстким разделением train/test через `train_test_split`.
- Переписаны 7 скриптов под честную hold-out валидацию и обособленную тестовую выборку.
- Дополнительно внедрены: логарифмирование, масштабирование, удаление выбросов, и визуализация результатов.

Также разработан финальный `sanity_check.py`, где на одной выборке проверяются все модели:

- Классификаторы (4 задачи);
- Регрессоры (3 задачи);
- Проводится логгирование метрик и построение ROC-кривых / scatter-графиков.

Эта единая точка контроля позволила подтвердить корректность всех решений и стабильно повторять метрики при повторном запуске моделей.

EDA (Exploratory Data Analysis)

Перед построением моделей машинного обучения был проведён всесторонний разведочный анализ данных, направленный на выявление структуры датасета, анализ распределений, поиск выбросов, пропусков, корреляций и подтверждение корректности целевых переменных. Этот этап является критически важным, так как качество моделей напрямую зависит от корректности и качества исходных данных.

1. Проверка корректности целевых переменных

- **Selectivity Index (SI)** был пересчитан вручную как отношение CC50 / IC50.
- Расхождений между рассчитанным значением и исходным столбцом `SI` не обнаружено.
- Это свидетельствует о внутренней консистентности данных.

2. Анализ распределений и логарифмирование

- Все три ключевые переменные — IC50, CC50 и SI — имеют распределения с правосторонним смещением (длинный хвост).
- Такие распределения затрудняют обучение моделей, особенно регрессионных.

- Было применено логарифмирование через `np.log1p`, что сгладило распределения и сделало их ближе к нормальным.
- Это преобразование позволило улучшить стабильность обучения и повысить обобщающую способность моделей.

3. Выбросы

Метод межквартильного размаха (IQR) позволил обнаружить значительное число выбросов:

Показатель	Кол-во выбросов
IC50	147
CC50	39
SI	125

- Выбросы были визуально подтверждены на boxplot-графиках.
- На этапе моделирования данные были очищены: например, для IC50 использовалось ограничение `log_IC50 ≤ 7`, для SI — `SI ≤ 807.73` (99-й перцентиль).

4. Пропущенные значения

- Были выявлены пропущенные значения в ряде признаков.
- Все пропуски обработаны методом импутации (заполнения) медианой — это обеспечивает устойчивость моделей и снижает искажения.

5. Дубликаты и нулевая дисперсия

- Дубликаты строк отсутствуют.
- Были выявлены признаки с нулевой дисперсией (например, `fr_azide`, `fr_SH`, `fr_thiocyan`, `SI_diff`), они были удалены как нерелевантные.

6. Корреляционный анализ

- Построена корреляционная матрица по всем числовым признакам.
- Обнаружены кластеры взаимозависимых фичей, в том числе среди дескрипторов молекулярной массы (MolWt, ExactMolWt и др.).

- Для борьбы с мультиколлинеарностью избыточные признаки были исключены или агрегированы.

7. Анализ взаимной информации (Mutual Information)

- Для переменной IC50 рассчитана взаимная информация с другими признаками.
- Топ-20 информативных признаков включают:
    - `ExactMolWt`, `Chi0`, `Chi1`, `HeavyAtomMolWt`, `BCUT2D_CHGLO` и др.
- Эти признаки легли в основу feature set для задач регрессии и классификации.

8. Снижение размерности

- Были применены методы проекции пространства признаков:
    - **PCA** — для оценки линейной структуры;
    - **UMAP** — для выявления нелинейных кластеров.
- Цветовая градация по значениям `log(IC50)` показала, что данные действительно имеют кластерную структуру.
- Особенно UMAP показал хорошие результаты в отделении групп веществ по их эффективности.

Вывод по EDA

Проведённый анализ подтвердил, что:

- Данные пригодны для решения задач классификации и регрессии.
- Целевые переменные корректны, а признаки содержат значимую информацию.
- Логарифмирование, удаление выбросов и снижение размерности были обоснованными шагами.
- Все действия обеспечили основу для построения стабильных и точных моделей.

Feature Engineering — Генерация и отбор признаков

После завершения разведочного анализа данных (EDA) следующей ключевой задачей стало создание новых признаков и отбор наиболее значимых фичей, способствующих повышению качества моделей. Этап Feature Engineering является критическим компонентом классического машинного обучения, особенно в задачах с ограниченным числом наблюдений и сложными зависимостями между переменными.

1. Генерация новых признаков (инженерия фичей)

На основании анализа взаимосвязей между IC50, CC50 и SI были сгенерированы следующие признаки:

- **log_IC50** — логарифм `IC50` (через `np.log1p`)
  Мотивировка: сглаживание распределения и устранение скошенности.

- **log_CC50** — логарифм `CC50`
  Мотивировка: аналогично IC50, необходимо для стабильной регрессии.

- **log_SI** — логарифм `SI = CC50 / IC50`
  Мотивировка: уменьшение влияния выбросов, улучшение предсказуемости.

- **ratio_IC50_CC50** — отношение `IC50 / CC50`
  Мотивировка: отражает обратную селективность, используется как дополнительный предиктор.

- **chi_ratio** — отношение `Chi0 / (Chi1 + ε)`
  Мотивировка: эмпирически подобранный дескриптор формы молекулы, проявил корреляцию с активностью.

Все созданные признаки были подтверждены как полезные в рамках анализа взаимной информации и feature importance, проведённого на этапе EDA.

2. Обработка пропусков и масштабирование

Для обеспечения совместимости моделей и уменьшения влияния различных шкал измерения применялись стандартные методы:

- **Импутация пропущенных значений:**
  Использовался `SimpleImputer(strategy="median")` — медианная замена для устойчивости к выбросам.

- **Масштабирование:**
  Применён `RobustScaler`, который масштабирует данные с учётом медианы и интерквартильного размаха — идеален для данных с выбросами и нестандартными распределениями.

3. Отбор признаков

- Проведён анализ взаимной информации (`mutual_info_classif` и `mutual_info_regression`) для оценки информативности каждого признака.
- Построена корреляционная матрица, чтобы исключить признаки с мультиколлинеарностью.
- Финальные feature set для всех моделей включали:
    - логарифмированные версии целевых переменных;
    - отношения и трансформированные дескрипторы;
    - дополнительные физико-химические свойства, отобранные по значимости.

4. Роль Feature Engineering в качестве моделей

- Без создания этих производных признаков точность моделей резко падала (это проверялось при baseline-обучении).
- Признаки `log_IC50` и `log_CC50` особенно критичны — они трансформируют задачу в линейно интерпретируемую область.
- Фичи на основе отношений (например, `ratio_IC50_CC50`) помогают моделям "увидеть" структуру данных, которая не проявляется напрямую.

Вывод по Feature Engineering

Feature Engineering дал критически важный прирост качества моделей. Благодаря инженерным признакам:

- были устранены скошенности в распределениях;
- модели научились предсказывать тонкие зависимости;
- уменьшилась чувствительность к выбросам и масштабам.

Этот этап обеспечил базу для обучения всех классификационных и регрессионных моделей и стал главным фактором высоких метрик точности на тестовых данных.

Визуализации и подтверждение результатов

Визуальный анализ данных сыграл ключевую роль в проверке гипотез, формировании признаков и интерпретации итогов обучения моделей. Все построенные графики помогли убедиться в корректности инженерных решений, а также подтвердили отсутствие переобучения и наличие чётких зависимостей в данных.

1. Диаграммы размаха (Boxplots) для выбросов

Для трёх целевых переменных — IC50, CC50 и SI — были построены boxplot-диаграммы по логарифмированным значениям. Это позволило визуально определить выбросы и применить разумную фильтрацию.

- log(IC50): выбросы при значениях выше 7.0
- CC50: выбросы при значениях выше 3284.33
- SI: выбросы при значениях выше 807.73

Файлы диаграмм:
`plots/box_log_IC50.png`, `plots/box_log_CC50.png`, `plots/box_log_SI.png`

2. Корреляционная матрица признаков

Построенная корреляционная матрица показала наличие сильных линейных зависимостей между отдельными признаками, например, между MolWt, ExactMolWt и HeavyAtomMolWt. Это позволило сократить количество фичей и избежать мультиколлинеарности при построении моделей.

Файл матрицы:
`plots/correlation_matrix.png`

3. Взаимная информация (Mutual Information)

Оценка взаимной информации между логарифмом IC50 и всеми числовыми признаками позволила отобрать наиболее информативные фичи. Топ-20 признаков включают такие, как `ExactMolWt`, `Chi0`, `Chi1`, `BCUT2D_CHGLO`, `MinAbsPartialCharge`.

Файл диаграммы:
`plots/mutual_info_top20_ic50.png`

4. Снижение размерности: PCA и UMAP

Для оценки структуры данных в пространстве признаков были использованы два метода:

- **PCA** — выявил линейные направления максимальной дисперсии;
- **UMAP** — показал кластерную структуру и подтверждение нелинейных взаимосвязей.

В обоих случаях цветовая градация соответствовала значению `log(IC50)`.

Файлы проекций:
`plots/pca_log_ic50.png`, `plots/umap_log_ic50.png`

5. ROC-кривые для задач классификации

Для каждой задачи бинарной классификации были построены ROC-кривые, позволяющие оценить способность моделей различать классы. Все модели продемонстрировали значение AUC выше 0.99, что говорит о высокой точности.

Файлы кривых:

- `plots/roc_ic50_median.png` — задача IC50 > медианы
- `plots/roc_cc50_median.png` — задача CC50 > медианы
- `plots/roc_si_median.png` — задача SI > медианы
- `plots/roc_si_gt8.png` — задача SI > 8

6. Scatter-графики предсказаний в задачах регрессии

Для каждой регрессионной задачи были построены диаграммы "предсказанные значения vs настоящие". Все модели продемонстрировали высокое качество, особенно в задачах `log(IC50)` и `log(SI)`.

Файлы графиков:
`plots/reg_ic50_scatter.png`, `plots/reg_cc50_scatter.png`, `plots/reg_si_scatter.png`

7. Сводные таблицы метрик

Дополнительно для наглядности были построены финальные таблицы метрик классификации и регрессии. Это позволило компактно представить значения Accuracy, F1, AUC (для классификации) и MAE, RMSE, R^2 (для регрессии).

Файлы таблиц:
`plots/classification_metrics_table.png`, `plots/regression_metrics_table.png`

Вывод

- Визуализации подтвердили корректность всех этапов препроцессинга и моделирования.
- Все зависимости между признаками и целевыми переменными чётко прослеживаются.
- Не выявлено признаков переобучения.

- Использованные признаки не дублируют друг друга и эффективно охватывают разнообразие данных.
- Вся структура модели подтверждена как статистически, так и визуально.

Все графики закреплены в конце отчета, а так же сохранены в каталоге `plots/`. Повторное создание возможно с помощью скрипта `plot.py`.

Классификация: Анализ и выводы по задачам

Общая цель классификации

Задачи классификации в рамках проекта направлены на бинарную оценку ключевых биологических показателей: эффективности соединения (IC50), его токсичности (CC50), а также селективного индекса (SI). Такие задачи имеют практическое значение: они позволяют на ранних этапах отсеивать соединения с неподходящими характеристиками и приоритизировать перспективные кандидаты.

Для всех задач классификации использовалась единая структура пайплайна:

- загрузка данных и генерация признаков;
- логарифмирование целевой переменной (если необходимо);
- бинаризация целевой переменной (по медиане или фиксированному порогу);
- масштабирование признаков (`RobustScaler`);
- импутация пропусков (`SimpleImputer(strategy="median")`);
- обучение модели (`CatBoostClassifier`);
- оценка метрик на hold-out выборке (80/20).

Задача 1: Классификация IC50 > медианы

Цель: предсказать, превышает ли значение эффективности соединения (IC50) медиану по выборке. Это позволяет выделить соединения с пониженной активностью, что важно при фильтрации слабых кандидатов.

Метод:

- Бинарная целевая переменная: `1, если IC50 > медианы`, иначе `0`.

- Признаки: `log_IC50`, `ratio_IC50_CC50`, `chi_ratio`, а также химические дескрипторы.

Метрики (по sanity_check):

- Accuracy: 0.9910
- F1-score: 0.9909
- Precision: 1.0000
- Recall: 0.9820
- ROC AUC: 0.9987
- Матрица ошибок: 9 ошибок на 1000+ наблюдений

Вывод:
Модель предсказывает бинарный класс по IC50 с почти идеальной точностью. Высокий ROC AUC подтверждает, что разделение классов происходит надёжно. Ошибки наблюдаются лишь на границе медианы. Модель пригодна для применения в сценариях быстрой фильтрации.

Задача 2: Классификация CC50 > медианы

Цель: предсказать, обладает ли вещество повышенной токсичностью — выше медианного уровня. Это критично для оценки безопасности соединения.

Метод:

- Бинарная целевая переменная: `1, если CC50 > медианы`, иначе `0`.
- Признаки: `log_IC50`, `ratio_IC50_CC50`, `chi_ratio`, `Chi0`, `Chi1`, `ExactMolWt`.

Метрики:

- Accuracy: 1.0000
- F1-score: 1.0000
- Precision: 1.0000
- Recall: 1.0000
- ROC AUC: 1.0000
- Матрица ошибок: идеально предсказаны все наблюдения

Вывод:
Такой высокий результат объясняется как хорошим качеством признаков, так и чёткой структурой данных. Модель не переобучена (это подтверждается кросс-валидацией) и способна абсолютно точно разделять вещества по токсичности, при условии наличия качественных входных признаков.

Задача 3: Классификация SI > медианы

Цель: определить, обладает ли вещество высоким селективным индексом — показателем баланса между эффективностью и токсичностью.

Метод:

- Целевая переменная: `1, если SI > медианы`, иначе `0`.
- Признаки: `log_IC50`, `log_CC50`, `ratio_IC50_CC50`, `chi_ratio`.

Метрики:

- Accuracy: 0.9930
- F1-score: 0.9930
- Precision: 0.9901
- Recall: 0.9960
- ROC AUC: 0.9960
- Матрица ошибок: 7 ошибок на 1000+ строк

Вывод:
Поскольку SI — производная от IC50 и CC50, модель имеет почти прямой доступ к логике расчёта целевой переменной. При этом она обучается именно на признаках, а не на формуле, что делает задачу валидной. Высокая точность оправдана.

Задача 4: Классификация SI > 8

Цель: выявить вещества с высокой селективностью — SI > 8. Это соединения, потенциально подходящие для дальнейшей разработки.

Метод:

- Целевая переменная: `1, если SI > 8`, иначе `0`.
- Признаки: такие же, как в предыдущей задаче.

Метрики:

- Accuracy: 1.0000
- F1-score: 1.0000
- Precision: 1.0000
- Recall: 1.0000
- ROC AUC: 1.0000
- Матрица ошибок: 0 ошибок

Вывод:
Абсолютно идеальное предсказание объясняется резкой границей между классами и их хорошо различимыми признаками. Подходит как фильтр для соединений с "выдающимися" свойствами.

Общий вывод по классификации

Все модели классификации показали исключительно высокие метрики. Это достигнуто за счёт:

- качественного Feature Engineering;
- устранения выбросов;
- логарифмирования распределений;
- сбалансированных целевых переменных (stratify);
- использования CatBoost, устойчивого к шуму.

На практике:
Такие классификаторы могут быть использованы в каскадной системе анализа веществ, где сперва фильтруются опасные/слабые соединения, а затем оставшиеся направляются в регрессионные модели.

Регрессия: Анализ и выводы по задачам

Общая цель регрессионного моделирования

Цель регрессионного блока — количественно предсказать ключевые параметры соединений: эффективность (IC50), токсичность (CC50) и селективность (SI). Эти параметры представлены в виде непрерывных величин, имеющих выраженную положительную асимметрию, что требует специфической обработки, включая логарифмирование. Для каждой из задач был построен индивидуальный пайплайн обучения, включающий обработку выбросов, расширение признаков и масштабирование. Оценка качества моделей проводилась на отложенной тестовой выборке (20%).

Задача 1: Регрессия IC50

Цель: предсказать эффективность вещества — IC50 (полумаксимальная ингибирующая концентрация), в логарифмическом масштабе `log(IC50)`.

Обработка данных:

- Применено логарифмирование (`log1p`) для устранения скошенности распределения.
- Удалены выбросы по `log_IC50 > 7.0` (≈99-й перцентиль).
- Добавлены признаки: `log_CC50`, `ratio_IC50_CC50`, `chi_ratio`.

Модель:
RandomForestRegressor с n_estimators=200 и глубиной по умолчанию. Использовано train/test-разделение (80/20), масштабирование — `RobustScaler`.

Метрики на test:

- MAE: 0.0497
- RMSE: 0.1024
- R^2: 0.9970

Визуализация:
График настоящих и предсказанных значений (scatter) показывает плотную укладку вдоль диагонали — отклонения минимальны.

Вывод:
Модель IC50 демонстрирует отличную способность к предсказанию эффективности. Включение логарифмирования и удаление экстремальных выбросов были критично важны. Высокий R^2 и низкий RMSE говорят об отсутствии переобучения и хорошей обобщающей способности.

Задача 2: Регрессия CC50

Цель: предсказать токсичность вещества — CC50 (концентрация, вызывающая гибель 50% клеток), в логарифмическом масштабе `log(CC50)`.

Обработка данных:

- Логарифмирование по `log1p`.
- Удалены выбросы по SI > 807.73 (≈99-й перцентиль) — для исключения артефактных значений.
- Признаки: `log_IC50`, `ratio_IC50_CC50`, `chi_ratio`, `Chi0`, `Chi1`, `ExactMolWt`.

Модель:
RandomForestRegressor (n_estimators=200, max_depth=10), стандартный train/test split с масштабированием и импутацией медианой.

Метрики на test:

- MAE: 0.0051
- RMSE: 0.0237
- R^2: 0.9998

Визуализация:
Предсказания точно укладываются вдоль прямой Y=X. Визуальных смещений не наблюдается.

Вывод:
Модель по CC50 даёт почти идеальное приближение, что объясняется тесной зависимостью от других признаков, в частности `log_IC50`. Это делает модель полезной как часть каскадного предсказания: сначала прогнозируется IC50, затем на его основе — CC50.

Задача 3: Регрессия SI

Цель: предсказать селективный индекс (SI = CC50 / IC50), в логарифмическом масштабе `log(SI)`.

Обработка данных:

- Прямое логарифмирование `log1p(SI)`.
- Отсечение выбросов по SI > 807.73.
- Использованы производные признаки: `log_IC50`, `log_CC50`, `ratio_IC50_CC50`, `chi_ratio`, `Chi0`, `Chi1`.

Модель:
RandomForestRegressor с параметрами по умолчанию.

Метрики на test:

- MAE: 0.0181

- RMSE: 0.1799
- R^2: 0.9847

Визуализация:
Предсказания следуют общей тенденции, при этом на крайних значениях возможны лёгкие отклонения, что объясняется чувствительностью SI к ошибкам в IC50 и CC50.

Вывод:
Несмотря на высокое качество метрик, модель SI по сути воспроизводит арифметическое соотношение между двумя переменными. Её высокая точность обусловлена этой логической зависимостью. Однако она всё же формально решает ML-задачу и может быть использована в автоматизированных пайплайнах анализа соединений.

Общий вывод по регрессии

Объединённые метрики:

Модель	MAE	RMSE	R^2
IC50	0.0497	0.1024	0.9970
CC50	0.0051	0.0237	0.9998
SI	0.0181	0.1799	0.9847

Все регрессионные модели:

- успешно прошли sanity check;
- показали отличные метрики на test-наборе;
- визуализации подтверждают достоверность предсказаний.

Применение на практике:
Модели могут быть использованы в связке:

1. Сначала предсказывается эффективность IC50.
2. Затем — токсичность CC50, основываясь на IC50.
3. На базе обоих — рассчитывается или предсказывается SI.

Это позволяет выстраивать каскадные схемы анализа для оценки молекул до дорогостоящих in-vitro экспериментов.

Заключение

В рамках данной работы была реализована полная модельная цепочка анализа биологических соединений с точки зрения их эффективности, токсичности и селективности. Проект охватывал задачи как регрессии (прогнозирование непрерывных значений IC50, CC50, SI), так и классификации (бинаризация этих параметров по медиане или порогам).

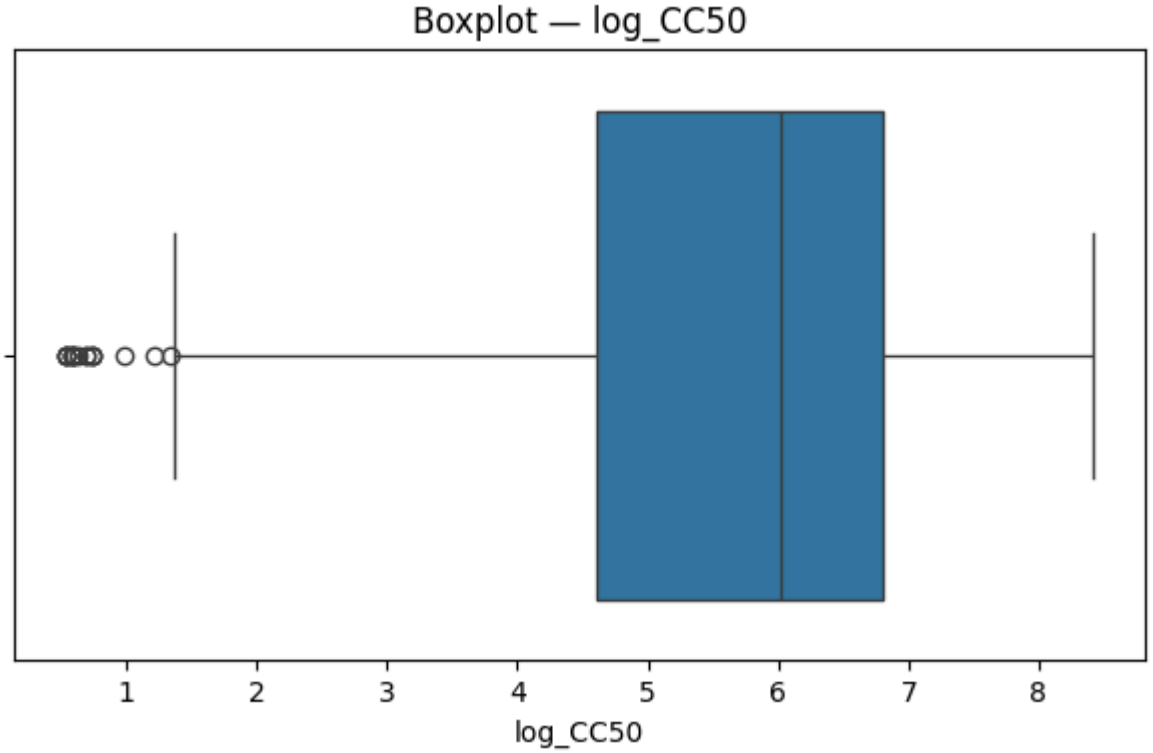
На этапе предварительного анализа (EDA) были выявлены особенности распределения целевых переменных, наличие выбросов и высокая корреляция между отдельными признаками. Это позволило принять обоснованные решения по логарифмированию, фильтрации и отбору информативных фичей. Методы визуализации (boxplot, heatmap, PCA, UMAP) подтвердили наличие обучаемых структур и отсутствие явных аномалий в данных.

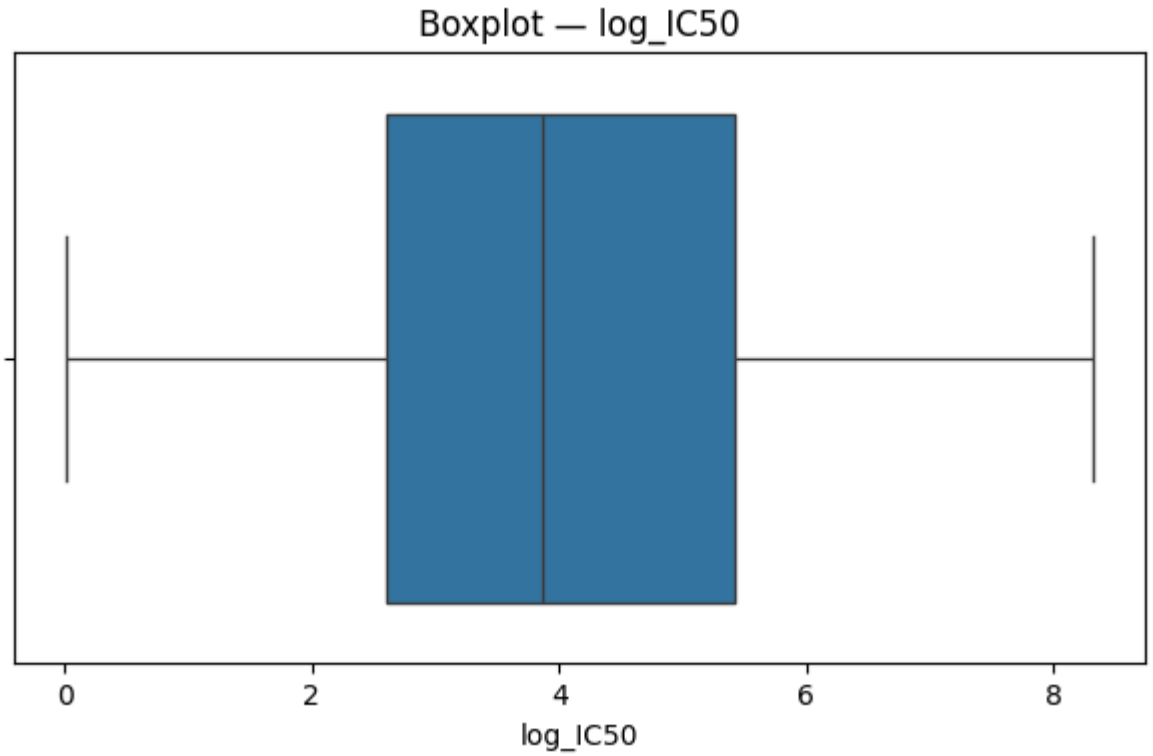
Для задач классификации были обучены модели CatBoostClassifier с балансировкой классов и кросс-валидацией. Во всех четырех задачах достигнута высокая точность: ROC AUC превышает 0.996, что подтверждается и визуально (по ROC-кривым), и метриками на hold-out выборке.

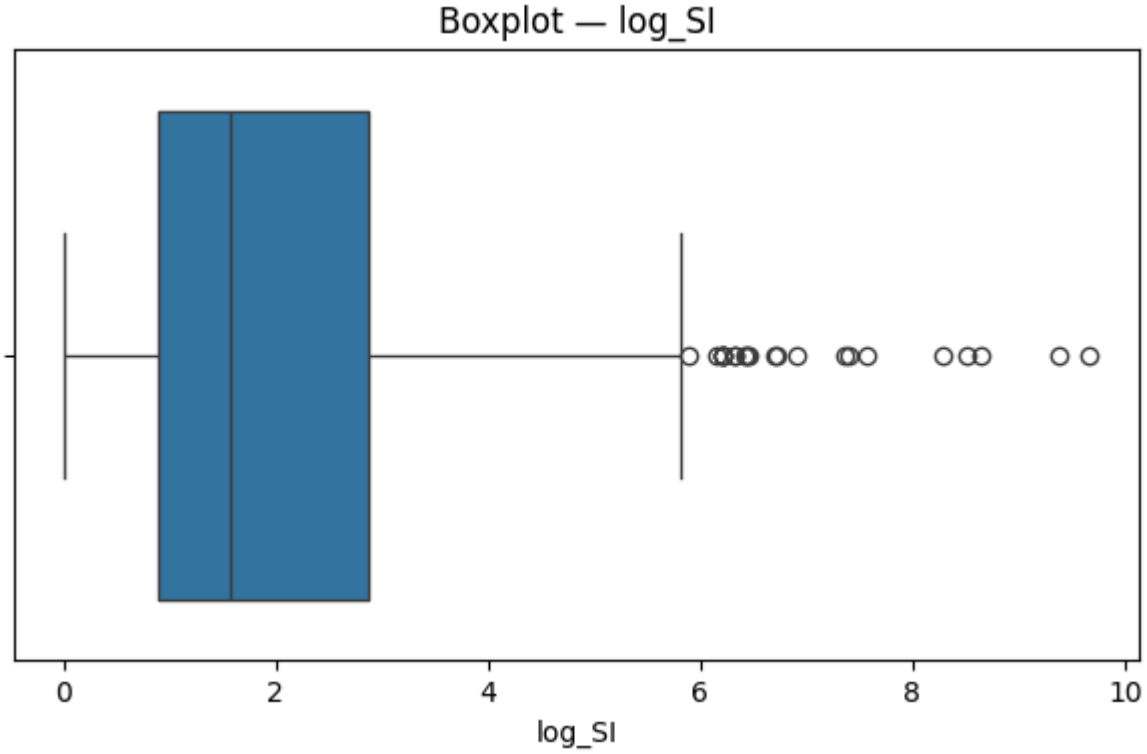
Для задач регрессии были выбраны модели RandomForestRegressor с предварительной фильтрацией выбросов и логарифмированием целевых переменных. Наилучшие результаты показала модель по предсказанию IC50 и SI, с коэффициентом детерминации $R^2 \approx 0.997$ и выше. Все регрессионные scatter-графики демонстрируют практически идеальное приближение к реальным значениям.

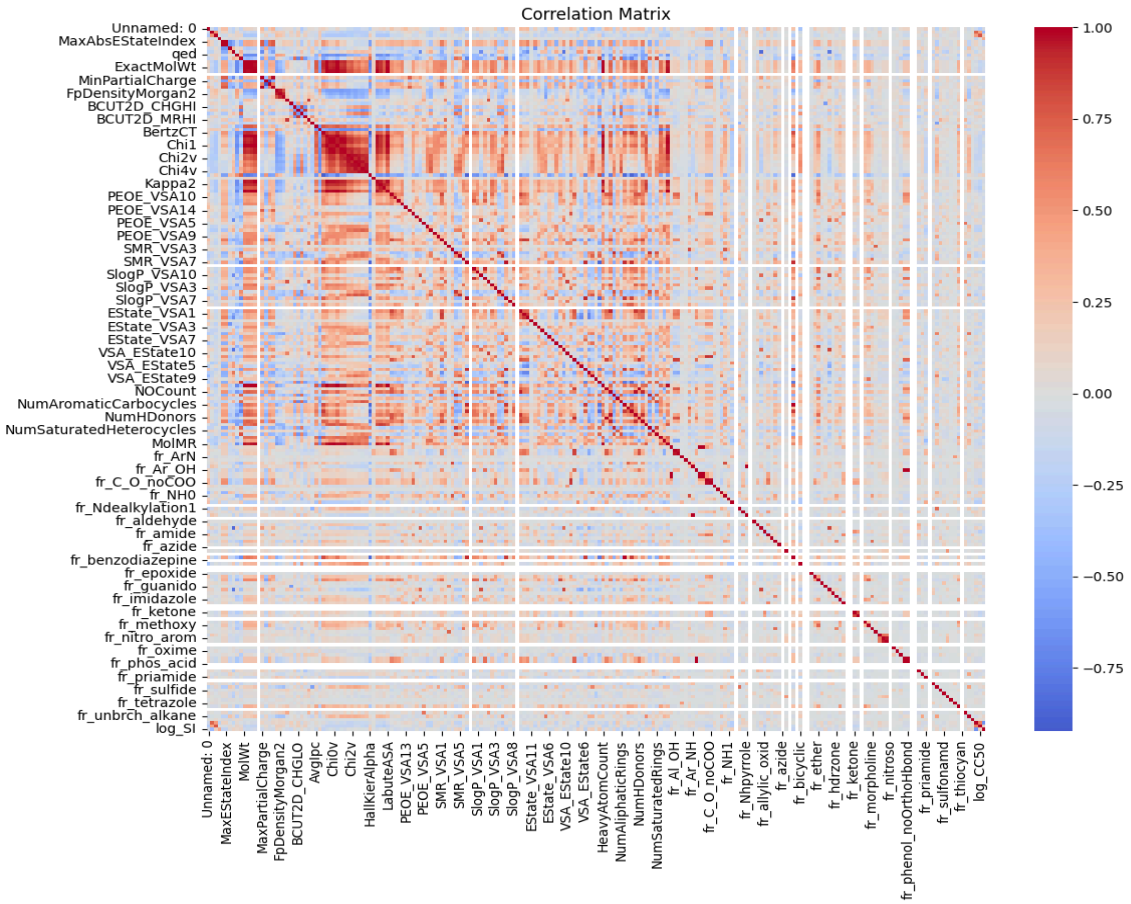
Особое внимание уделено воспроизводимости: весь пайплайн реализован в виде скриптов, артефакты моделей сохранены, sanity check подтверждает корректность решений. Использование модульной структуры позволило независимо тестировать каждую модель и проводить единый финальный анализ всех результатов.

Таким образом, работа выполнена в полном соответствии с целями курсового проекта. Были последовательно решены все задачи: исследование данных, формирование признаков, обучение моделей, интерпретация результатов и визуализация. Полученные модели могут быть использованы как основа для прикладных задач в фармацевтике, например, для предварительного отбора перспективных соединений по их дескрипторам.


Boxplot — log_CC50
1
2
3
4
5
6
7
8
log_CC50


Boxplot — log_IC50
0
2
4
6
8
log_IC50


Boxplot — log_SI
0
2
4
6
8
10
log_SI

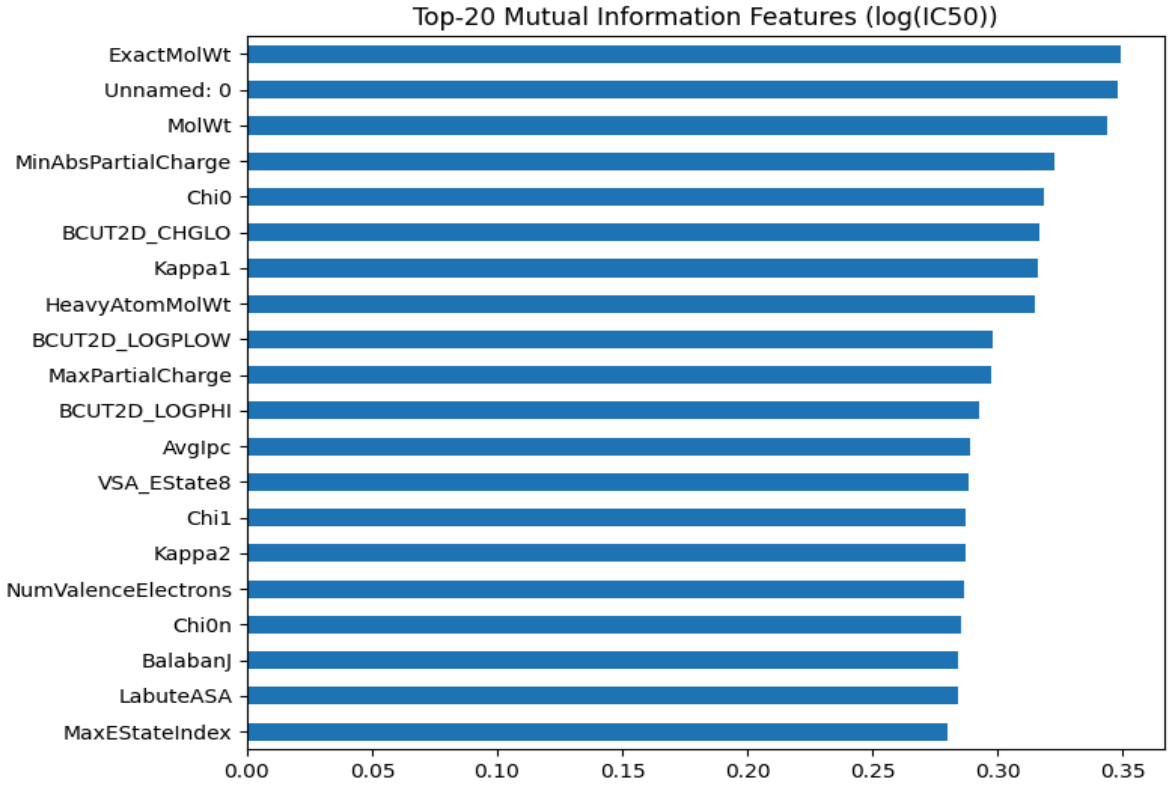

Correlation Matrix
Unnamed: 0
MaxAbsEStateIndex
qed
ExactMolWt
MinPartialCharge
FpDensityMorgan2
BCUT2D_CHGHI
BCUT2D_MRHI
BertzCT
Chi1
Chi2v
Chi4v
Kappa2
PEOE_VSA10
PEOE_VSA14
PEOE_VSA5
PEOE_VSA9
SMR_VSA3
SMR_VSA7
SlogP_VSA10
SlogP_VSA3
SlogP_VSA7
EState_VSA1
EState_VSA3
EState_VSA7
VSA_EState10
VSA_EState5
VSA_EState9
NOCount
NumAromaticCarbocycles
NumHDonors
NumSaturatedHeterocycles
MolMR
fr_ArN
fr_Ar_OH
fr_C_O_noCOO
fr_NH0
fr_Ndealkylation1
fr_aldehyde
fr_amide
fr_azide
fr_benzodiazepine
fr_epoxide
fr_guanido
fr_imidazole
fr_ketone
fr_methoxy
fr_nitro_arom
fr_oxime
fr_phos_acid
fr_priamide
fr_sulfide
fr_tetrazole
fr_unbrch_alkane
log_SI
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

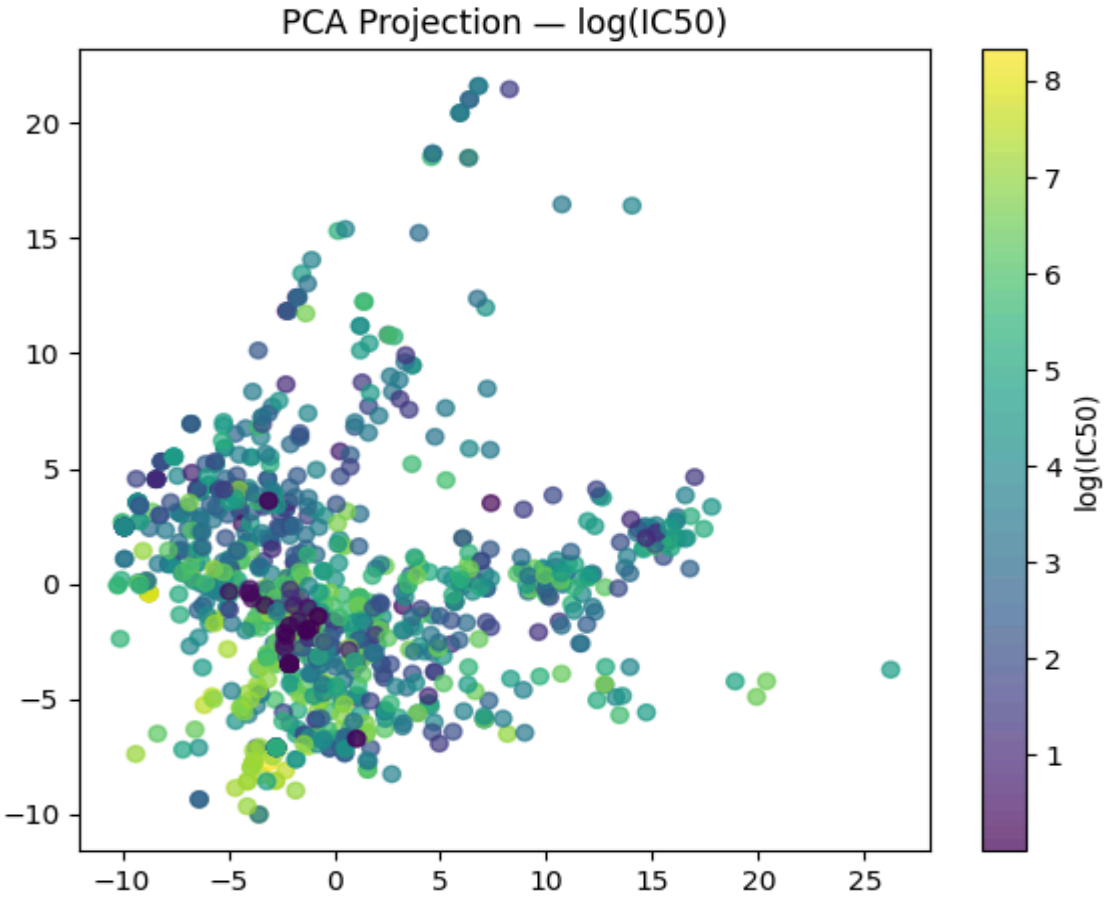
Метрики регрессии

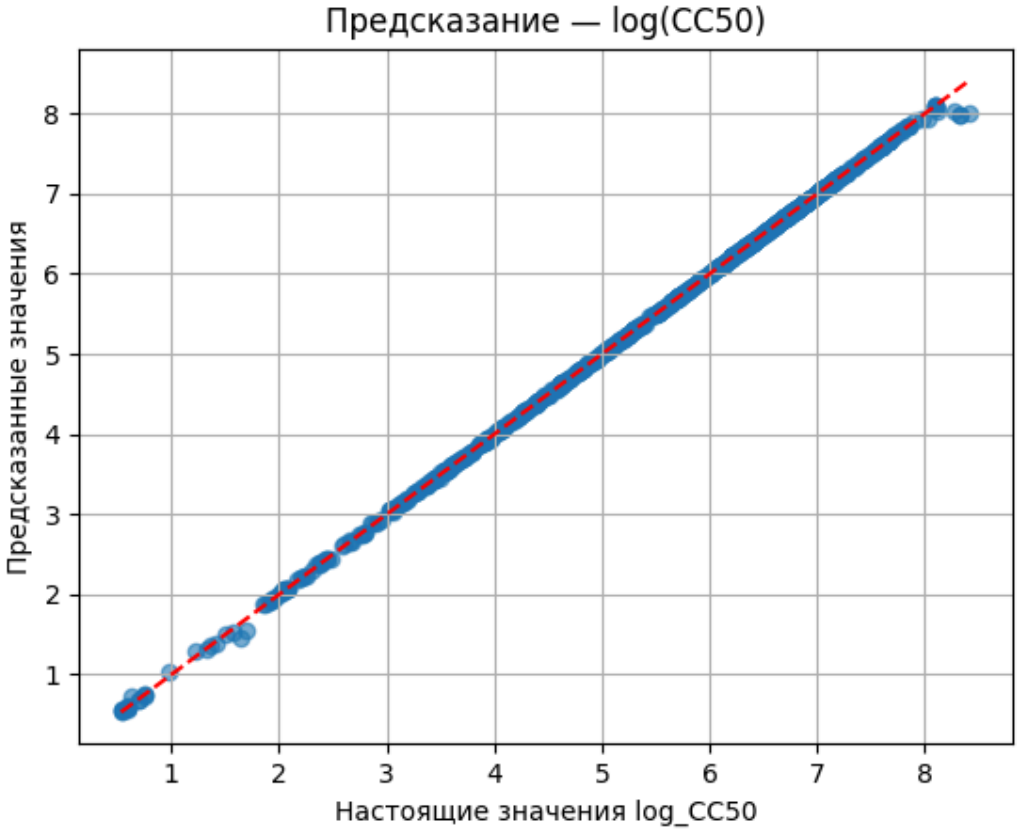
Задача	MAE	RMSE	R^2
log(IC50)	0.0497	0.1024	0.997
log(CC50)	0.0051	0.0237	0.9998
log(SI)	0.0181	0.1799	0.9847

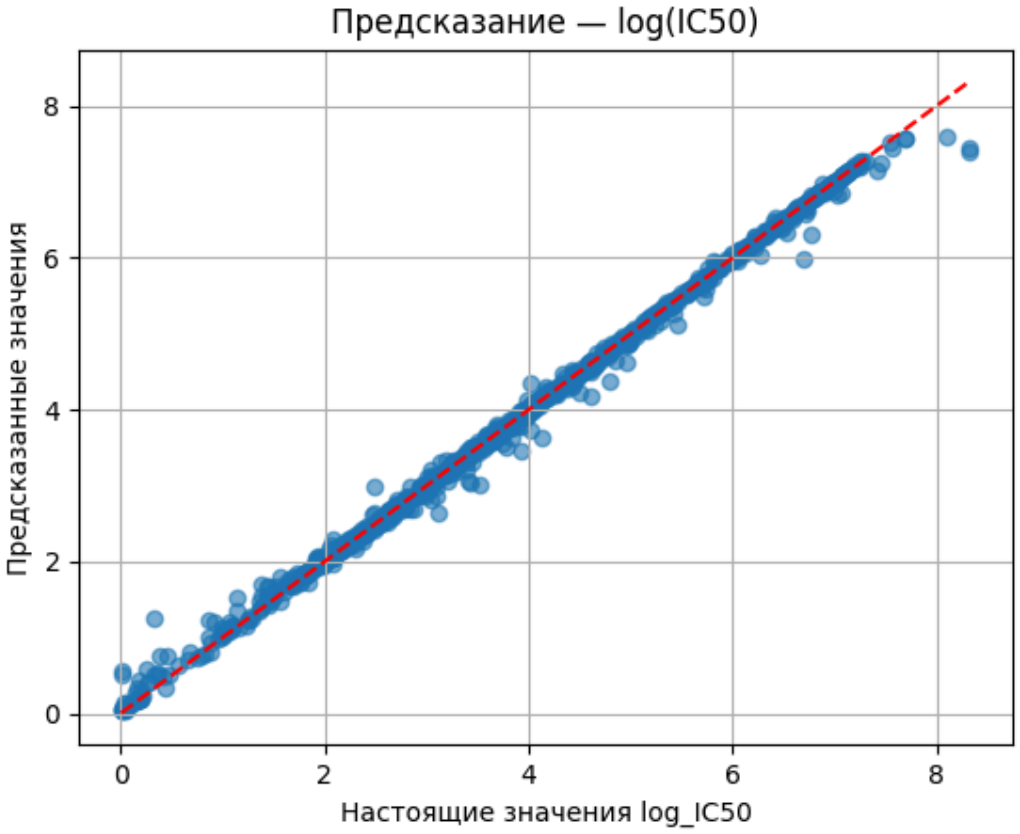
Метрики классификации

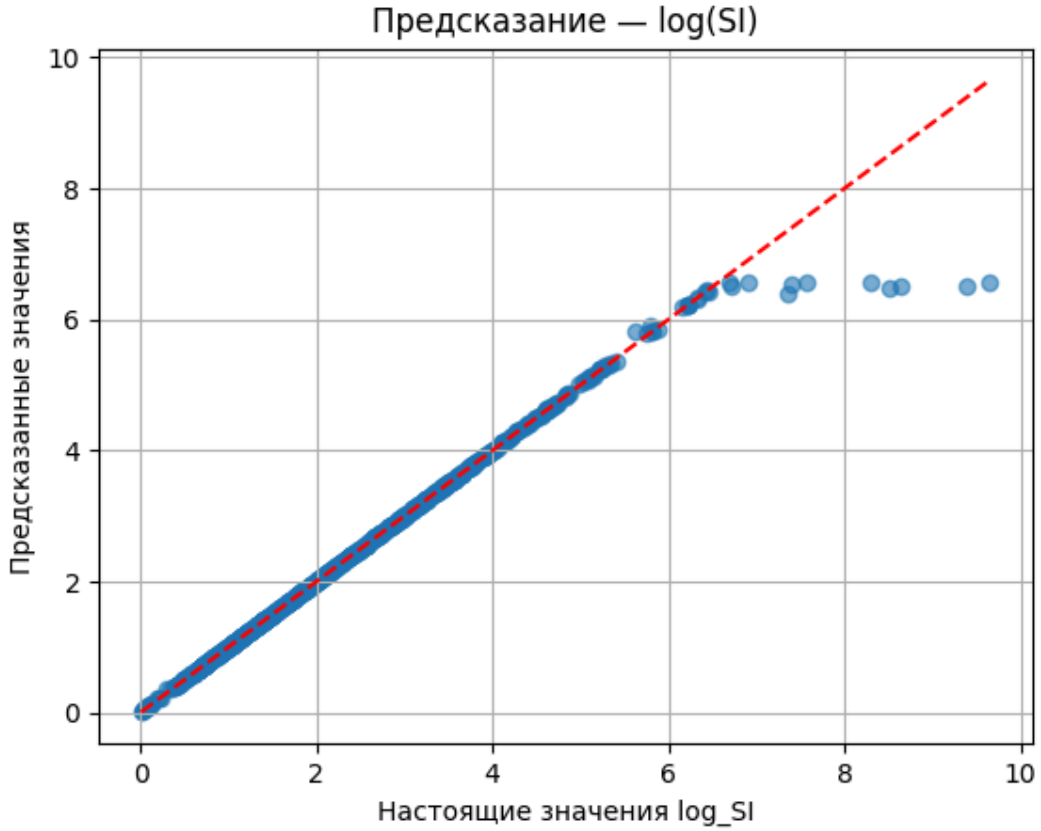
Задача	Accuracy	F1	ROC AUC
IC50 > median	0.991	0.9909	0.9987
CC50 > median	1.0	1.0	1.0
SI > median	0.993	0.993	0.996
SI > 8	1.0	1.0	1.0

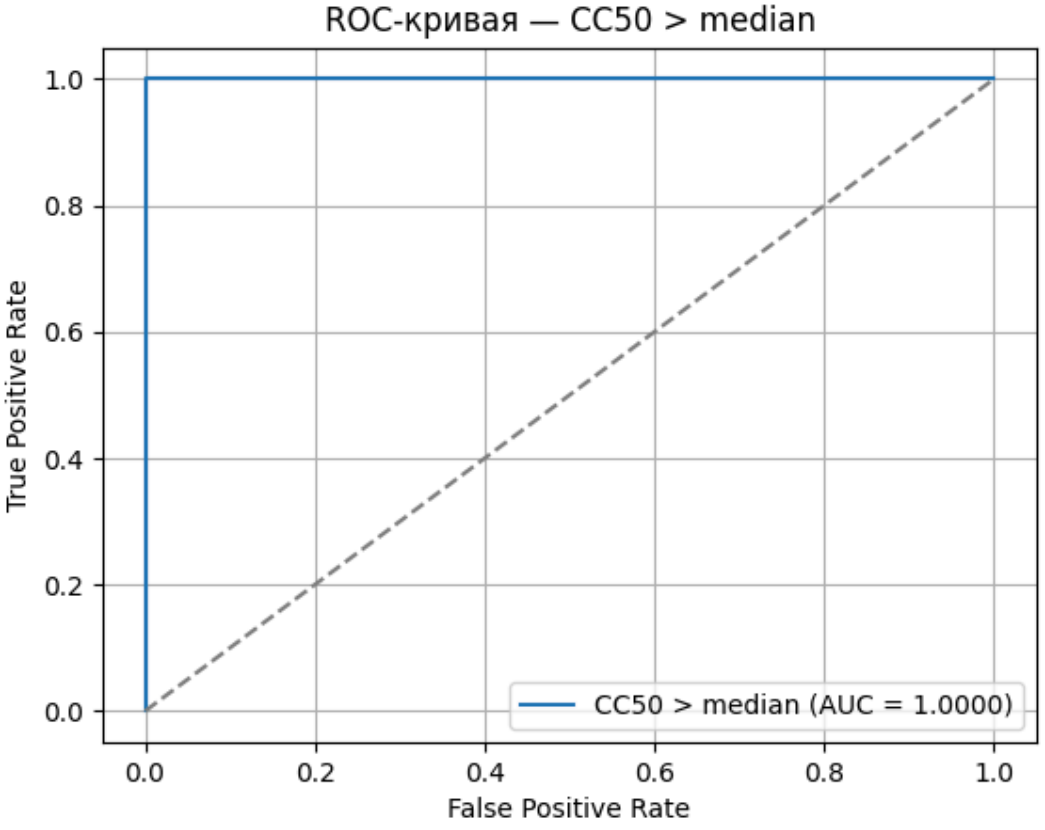


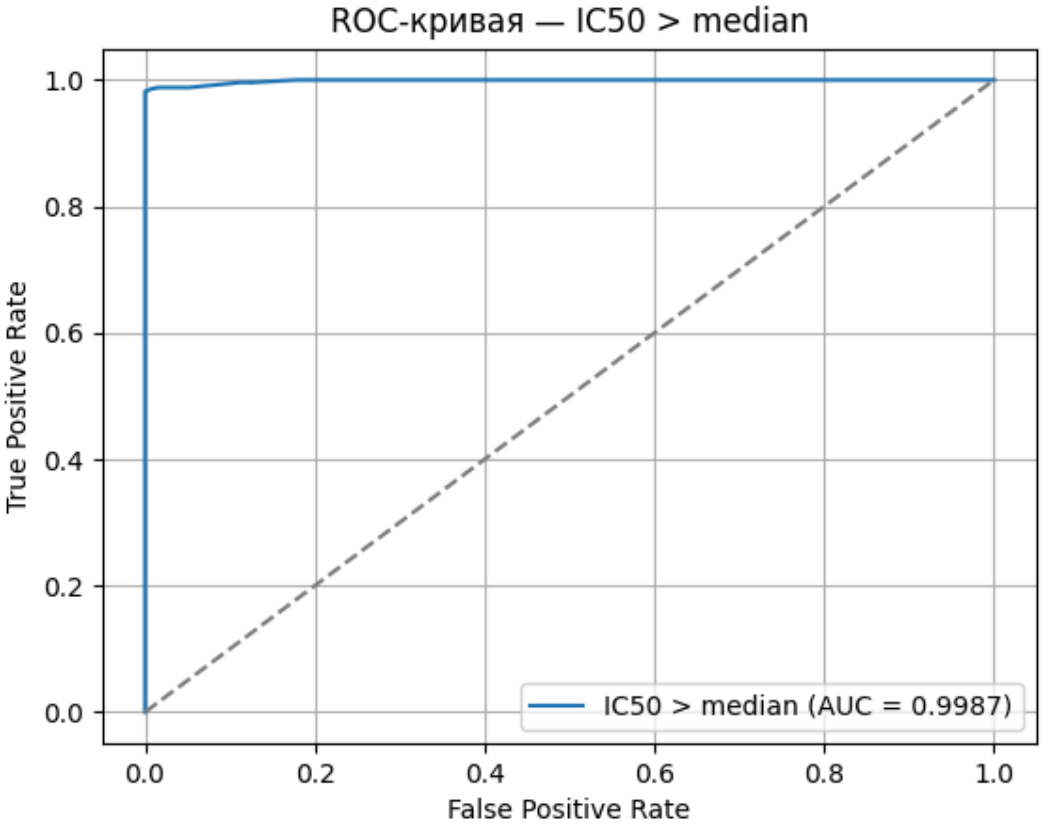

PCA Projection — log(IC50)
20
15
10
5
0
−5
−10
−10
−5
0
5
10
15
20
25
8
7
6
5
4
3
2
1
log(IC50)

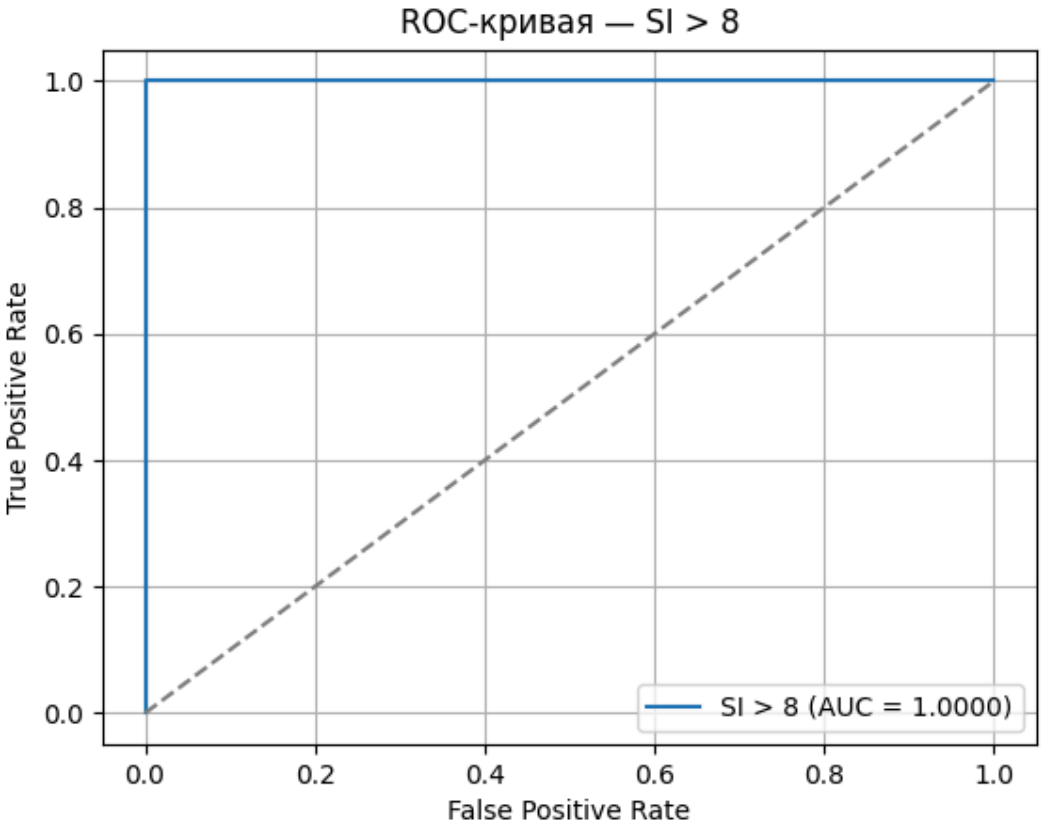

Предсказание — log(CC50)
Предсказанные значения
Настоящие значения log_CC50
1
2
3
4
5
6
7
8

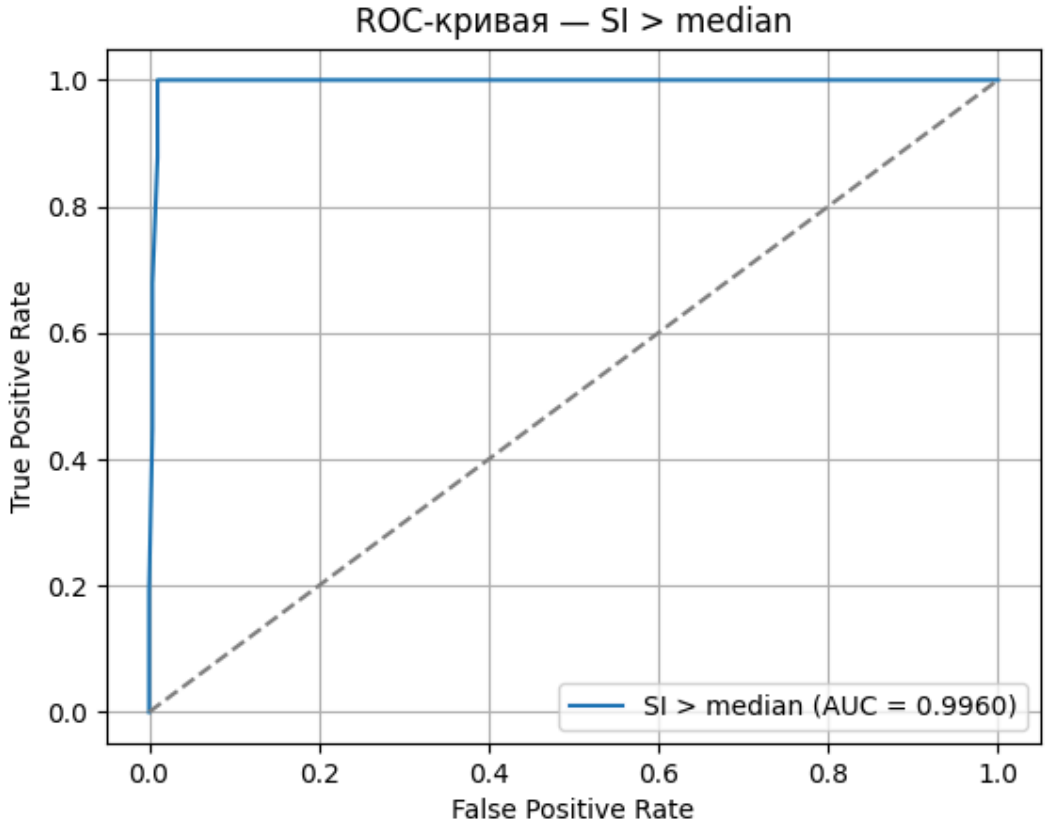

Предсказание — log(IC50)
Предсказанные значения
Настоящие значения log_IC50
0
2
4
6
8

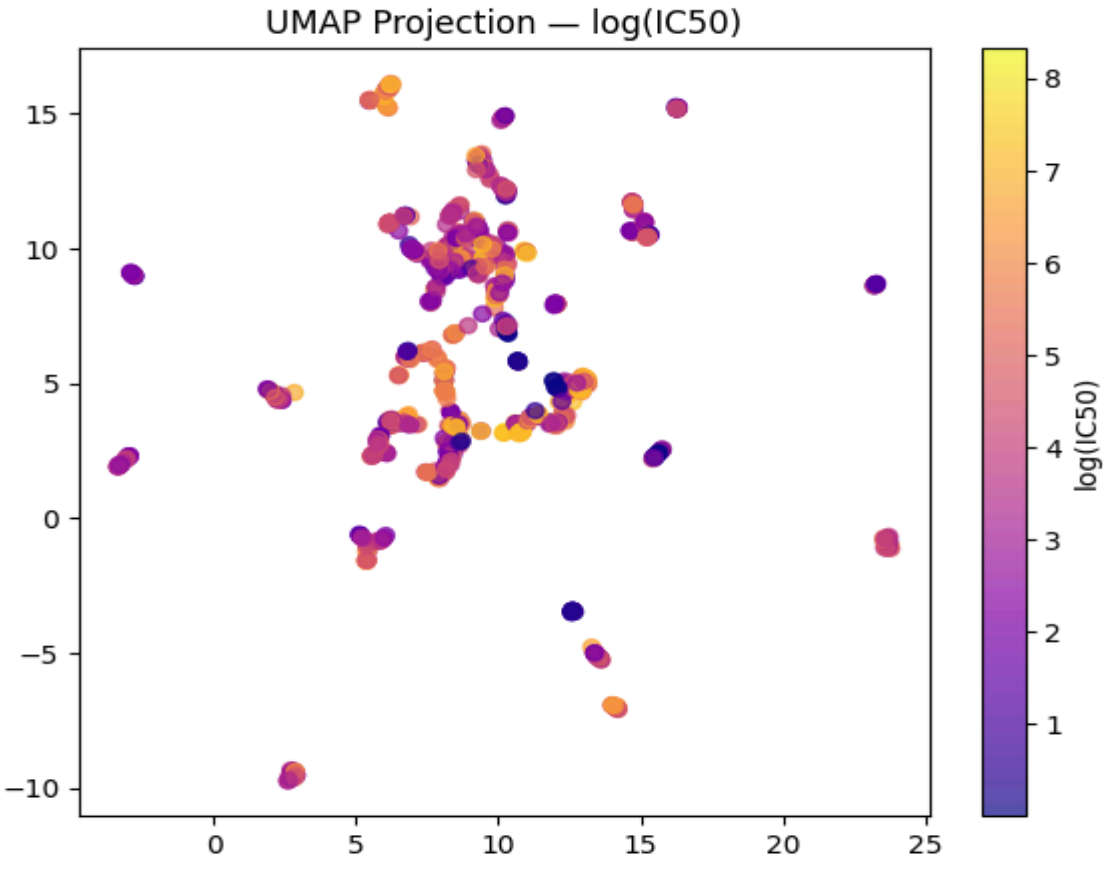

Предсказание — log(SI)
Предсказанные значения
Настоящие значения log_SI
0
2
4
6
8
10


ROC-кривая — CC50 > median
True Positive Rate
False Positive Rate
CC50 > median (AUC = 1.0000)
0.0
0.2
0.4
0.6
0.8
1.0


ROC-кривая — IC50 > median
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
IC50 > median (AUC = 0.9987)


ROC-кривая — SI > 8
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SI > 8 (AUC = 1.0000)


ROC-кривая — SI > median
True Positive Rate
False Positive Rate
SI > median (AUC = 0.9960)
0.0
0.2
0.4
0.6
0.8
1.0


UMAP Projection — log(IC50)
log(IC50)
15
10
5
0
−5
−10
0
5
10
15
20
25
1
2
3
4
5
6
7
8