**Вывод по курсовой работе**

**Цель исследования**: построить прогностические модели, которые помогут при разработке лекарственных препаратов, на основе предоставленных данных различных химических соединений и показателей IC₅₀, CC₅₀ и SI – эффективности и токсичности веществ.

**Актуальность исследования**: способы упростить и ускорить процесс создания лекарств востребованы в настоящее время, так как часто специалистам приходится разрабатывать новые препараты в короткие сроки. Также подобные исследования важны из-за того, что помогают сфокусироваться на более перспективных данных с помощью отсеивания нерелевантных и неэффективных признаков.

**Задачи исследования**: подготовить предварительный анализ данных (EDA), обработать признаки должным образом, применить Feature Selection для выбора наиболее коррелирующих признаков, протестировать несколько алгоритмов машинного обучения для поиска наилучшего, применить выбранный алгоритм для выполнения задач по регрессии (IC₅₀, CC₅₀ и SI) и классификации (IC₅₀ > median, CC₅₀ > median и SI > median и SI > 8), выбрать самые важные признаки для успешного проектирования лекарств.

**Метрики для оценки качества построенных моделей**: MAE, MSE, R2 для задач регрессии, Precision, Accuracy, Recall, F1\_score для классификации.

**Использованные в процессе выполнения библиотеки**: numpy, pandas, matplotlib, seaborn, sklearn, xgboost.

**Анализ датасета**: размерность данных составляет (1001, 214), где 214 – число признаков, 1001 – число наблюдений. В нем содержатся различные признаки, описывающие химико-физические, структурные и другие свойства предоставленных химических соединений. В данных присутствуют целочисленные (int) и дробные (float) показатели. Целевые переменные - IC₅₀, CC₅₀ и SI – показывают концентрации соединений для подавления вирусной активности на 50% (IC₅₀), вещество, вызывающее гибель 50% клеток, (CC₅₀) и индекс селективности (SI). В данных присутствуют малоинформативные признаки, выбросы, а также низкокоррелирующие признаки.

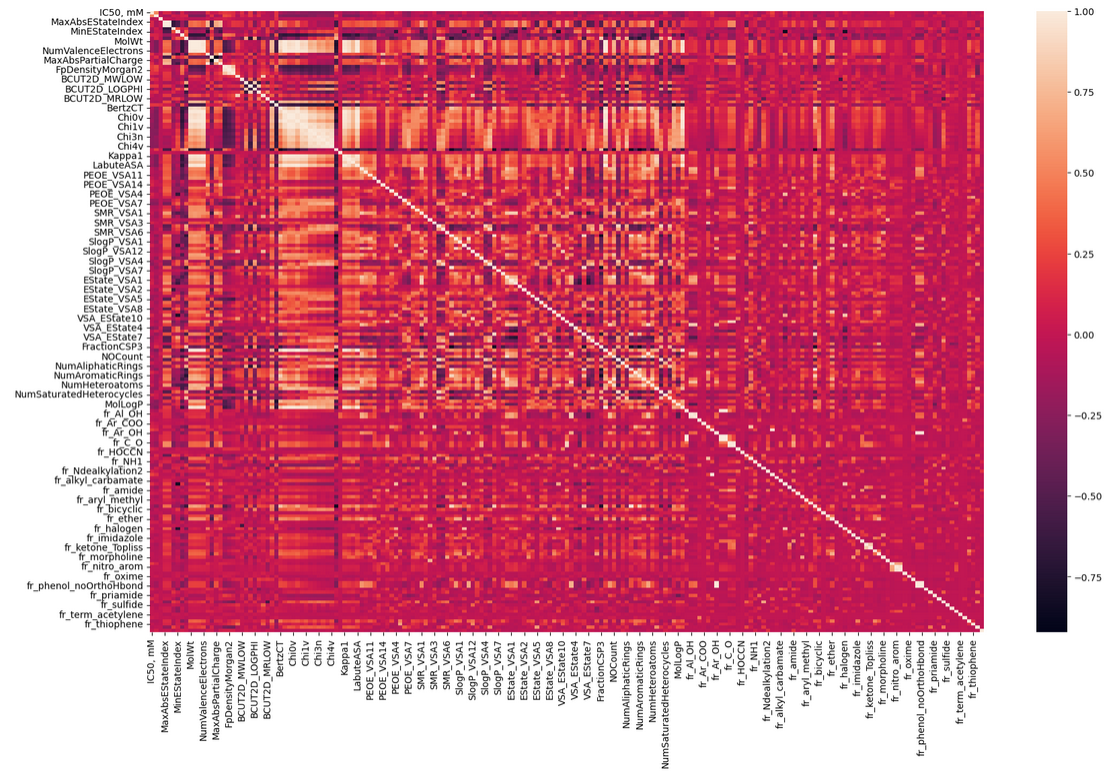


Рисунок 1. Корреляционная матрица

**Предварительный анализ данны**х: в качестве EDA были проведены следующие действия:

1. Проверка и удаление на дубликаты (с помощью .drop\_duplicates();
2. Удаление малоинформативных записей с помощью .nunique() и .drop();
3. Удаление неинформативных признаков ('Unnamed: 0');
4. Визуализация выбросов в целевых переменных с помощью .boxplot() и их удаление с помощью квантилей (Q1/Q3 +- 1.5 \* IQR\_ic);
5. Проведение логарифмирования целевых переменных.

Целевые переменные были сильно разбросаны в значениях, из-за чего появилась необходимость их логарифмировать.

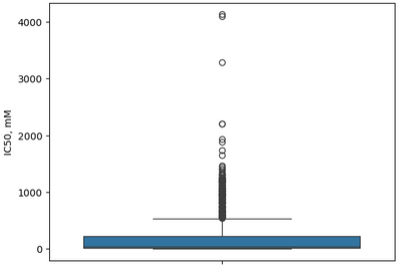


Рисунок 2. Выбросы IC50

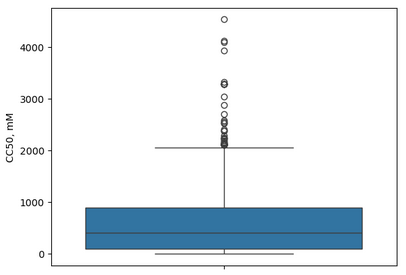


Рисунок 3. Выбросы CC50

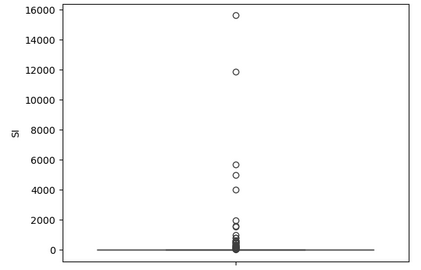


Рисунок 4. Выбросы SI

1. До логарифмирования задавалось значение SI > 8 в отдельную переменную для последующей задачи классификации SI > 8;
2. Логарифмирование целевых переменных IC₅₀, CC₅₀ и SI;
3. Первичное обозначение медианы до разделения данных;
4. Разделение данных на тренировочную и тестовую выборки по каждой из целевых переменных для каждой из задач регрессии и классификации;
5. Заполнение пропусков в каждой выборке средним арифметическим;
6. Расчет медианы по каждой из тренировочных выборок, обновление train выборок для задач классификации по каждой из медиан и SI > 8, применение полученных значений к test выборке. Это было сделано ради того, чтобы избежать завышения метрик и data leakage;
7. С помощью SelectKbest, mutual\_info\_regression и VarianceThreshold (threshold=0.01) выбор 50 наиболее важных для каждой целевой переменной признаков, действия проводились на train выборках, после чего соответствующие им test выборки проходили те же изменения.
8. Стандартизация всех train и test выборок с помощью StandardScaler (всего получилось 7 пар X\_train и X\_test).

**Подбор алгоритмов МО для решения поставленных задач** **– регрессия**: в качестве исследования тестировались различные алгоритмы для поиска лучшей метрики R2, которая также обозначала бы, что данный алгоритм подходит для анализа особенностей данных, полученных в химических лабораториях, и учитывает их молекулярные и структурные особенности.

Использовались наиболее популярные алгоритмы регрессии из библиотеки sklearn.

Для задач регрессии сначала было протестировано использование исключительно алгоритмов машинного обучения с учителем (линейная регрессия, случайный лес, градиентный бустинг, XGBregressor) на IC₅₀, однако метрика R2-score была низкой во всех алгоритмах.

|  |  |
| --- | --- |
| **Алгоритм** | **R2\_score (на тесте)** |
| Линейная регрессия | 0.3403781793341438 |
| Случайный лес | 0.39951600346802496 |
| Градиентный бустинг | 0.4397512398595178 |

Вследствие этого было выдвинуто предположения использовать кластеризацию на каждом X-датасете по целевой переменной. С помощью кластеризации удалось выявить, что данные плохо поддаются разделению на кластеры, из-за чего предположение об эффективности данного метода была опровергнуто (метрики представлены ниже).

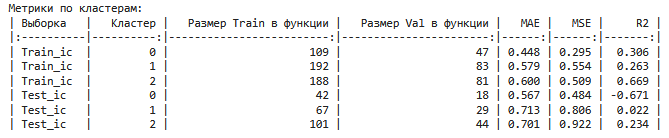


Рисунок 5. Кластеризация и случайный лес

Было принято решение использовать только алгоритмы машинного обучения с учителем. Для регрессии IC₅₀ наиболее эффективным алгоритмом стал градиентный бустинг. Для подбора наилучших гиперпараметров данного алгоритма был использован RandomizedSearchCV из библиотеки sklearn. Лучшими параметрами стали: {'learning\_rate': 0.11753971856328177, 'max\_depth': 3, 'n\_estimators': 77}, метрика R2\_score на тесте дала 0.4276572363509815.

Для регрессии CC₅₀ наиболее эффективным также оказался алгоритм градиентного бустинга, лучшие гиперпараметры и R2\_score оказались следующими: {'learning\_rate': 0.05680559213273095, 'max\_depth': 5, 'n\_estimators': 264}, R2 на тесте: 0.32046574376882486.

Для регрессии SI наиболее эффективным оказался алгоритм случайного леса, лучшие гиперпараметры и R2\_score оказались следующими: Лучшие параметры: {'max\_depth': 6, 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 97}, R2 на тесте: 0.25915523203691504.

**Подбор алгоритмов МО для решения поставленных задач** **– классификация:** в задачах классификации были также протестированы различные алгоритмы – логистическая регрессия, метод k ближайших соседей, случайный лес, XGBClassifier. В этот раз метрика accuracy была изначально не слишком плохой – 0.6378737541528239, 0.6877076411960132, 0.6910299003322259 и 0.6744186046511628 соответственно. Для того, чтобы самый успешный алгоритм имел лучшие гиперпараметры, также использовался RandomizedSearchCV. Использование классического GridSearch оказалось непроизводительным, так как даже с уменьшенным количеством признаков подбор гиперпараметров с помощью данного метода занимал много времени.

Для классификации IC₅₀ > своей медианы наиболее эффективным оказался алгоритм случайного леса (classifier). Метрики на тесте получились следующие: Accuracy на тесте: 0.707641196013289; F1 на тесте: 0.7124183006535948; Precision на тесте: 0.68125; Recall на тесте: 0.7465753424657534 с гиперпараметрами {'max\_depth': 9, 'min\_samples\_leaf': 3, 'min\_samples\_split': 15, 'n\_estimators': 130}.

Для классификации CC₅₀ > своей медианы наиболее эффективным также оказался алгоритм случайного леса (classifier). Метрики на тесте получились следующие: Accuracy на тесте: 0.760797342192691; F1 на тесте: 0.7721518987341772; Precision: 0.7672955974842768; Recall: 0.7770700636942676 с гиперпараметрами {'max\_depth': 9, 'min\_samples\_leaf': 3, 'min\_samples\_split': 8, 'n\_estimators': 290}.

Для классификации SI > своей медианы наиболее эффективным также оказался алгоритм случайного леса (classifier). Метрики на тесте получились следующие: Accuracy на тесте: 0.6578073089700996; F1 на тесте: 0.6199261992619927; Precision: 0.7; Recall на тесте: 0.5562913907284768 с гиперпараметрами {'max\_depth': 9, 'min\_samples\_leaf': 3, 'min\_samples\_split': 8, 'n\_estimators': 290}.

Для классификации SI > 8 наиболее эффективным также оказался алгоритм случайного леса (classifier). Метрики на тесте получились следующие: Accuracy на тесте: 0.9534883720930233; F1 на тесте: 0.9761092150170648; Precision: 0.959731543624161; Recall на тесте: 0.9930555555555556 с гиперпараметрами {'max\_depth': 8, 'min\_samples\_leaf': 2, 'min\_samples\_split': 13, 'n\_estimators': 211}.

Таким образом, для работы с химическими соединениями и оценки показателей IC₅₀, CC₅₀ и SI больше всего подходят алгоритмы градиентного бустинга и случайного леса (regressor) для регрессии и случайного леса (classifier) для классификации. Таким образом, можно предположить, что в данных подобного вида есть нелинейные зависимости, так как данные алгоритмы учитывают сложные взаимосвязи между признаками, в том числе между их комбинациями. Также стоит отметить, что деревья помогают учитывать взаимодействие признаков. Более того, в предоставленных данных изначально было ненормальное распределение и много шума, с чем бустинг и деревья хорошо справляются, так как они не требуют предположения о нормальности распределения и устойчивы к шуму.

**Общий вывод по проделанной работе и визуализация признаков**: для иллюстрации сделанных выводов выведем список 25 наиболее важных признаков в работе выбранных алгоритмов по каждой из задач. Для визуализации данных признаков использовался .feature\_importances\_.

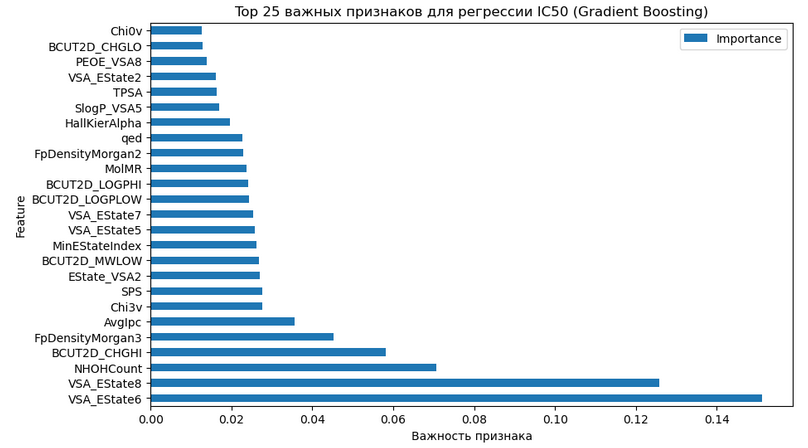


Рисунок 6. Top 25 важных признаков для регрессии IC50

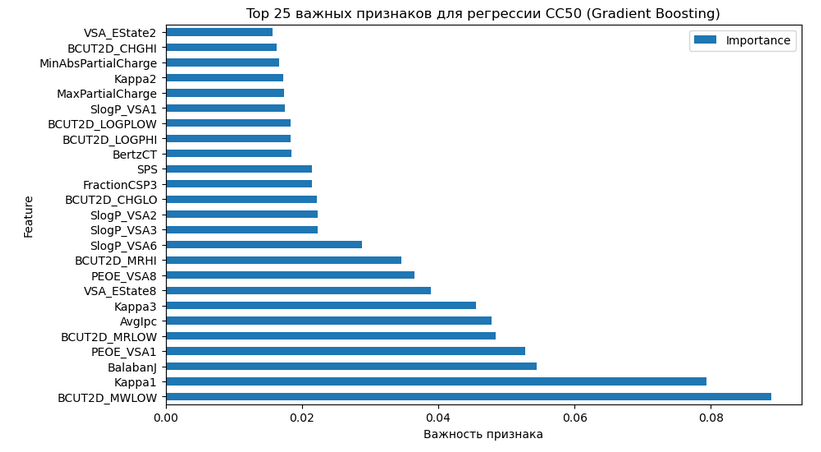


Рисунок 7. Top 25 важных признаков для регрессии CC50

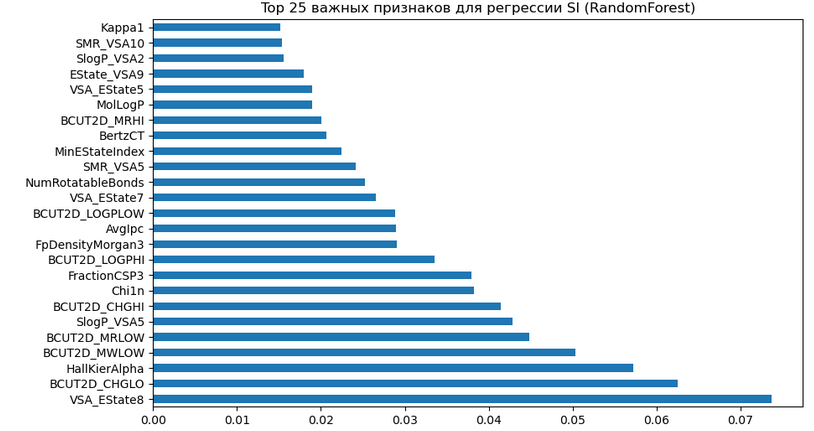


Рисунок 8. Top 25 важных признаков для регрессии SI

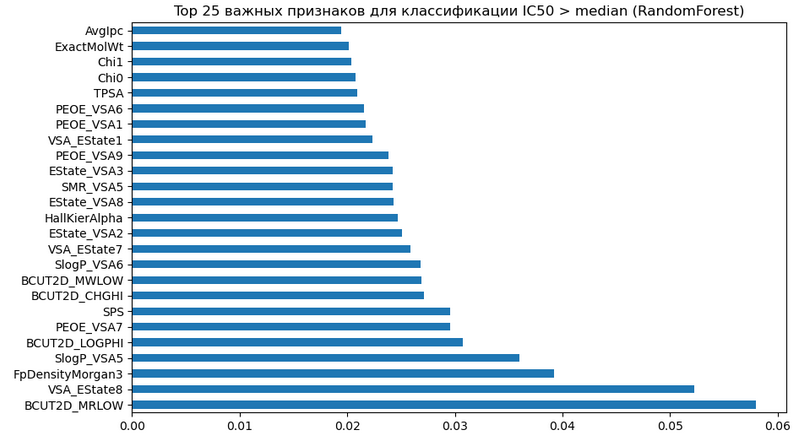


Рисунок 9. Top 25 важных признаков для классификации IC50 > median

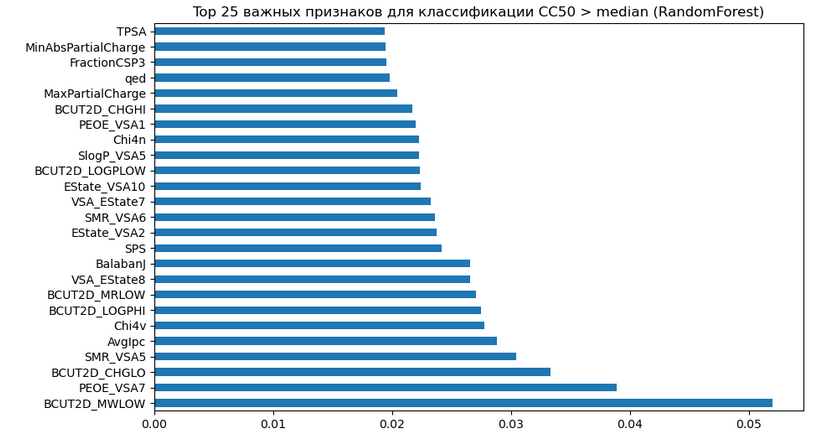


Рисунок 10. Top 25 важных признаков для классификации CC50 > median

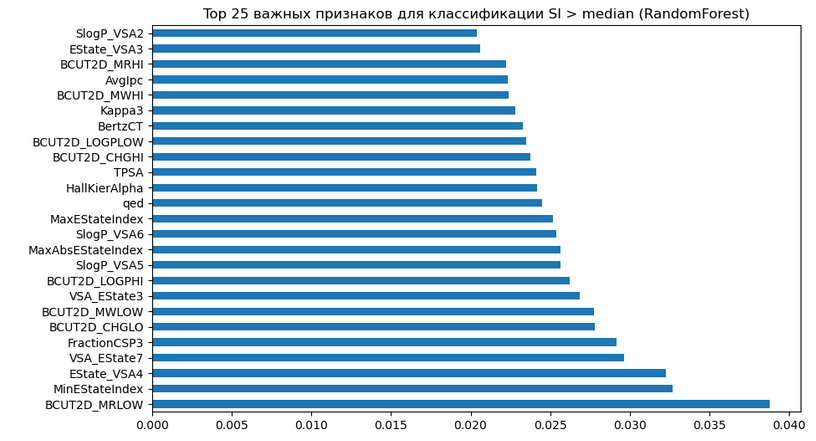


Рисунок 11. Top 25 важных признаков для классификации SI > median



Рисунок 12. Top 25 важных признаков для классификации SI > 8

Признаки, визуализированные на каждом из рисунков, являются наиболее информативными и важными для каждой из задач в отдельности. Это является ценной исследовательской информацией, которая может быть использована в будущих исследованиях как отправная точка в зависимости от желаемого результата.

Примененные модели регрессии позволяют проанализировать эффективность и токсичность соединений еще до их синтеза, а классификация позволяет оценить, пригодны ли созданные лекарства для реализации на потребителей. Применение методов машинного обучения позволяет ускорить анализ соединений, сделать его более эффективным с помощью концентрации на более «успешных» признаках и построить аналогии на другие соединения с помощью результатов, полученных в уже реализованных алгоритмах, в похожих задачах.