# ВВЕДЕНИЕ

Создание нового лекарственного препарата представляет собой сложный и многоступенчатый процесс, включающий определение химической формулы вещества, его синтез, проведение первичных биологических испытаний и последующее клиническое тестирование. Каждый из этих этапов требует значительных временных и ресурсных затрат. На помощь приходят современные методы машинного обучения, способные значительно ускорить разработку лекарств.

Например, с помощью различных алгоритмов можно прогнозировать биологическую активность соединений и выделять наиболее перспективные комбинации параметров для дальнейших исследований. Однако успешная реализация таких проектов возможна лишь при условии тесного и эффективного взаимодействия между специалистами в области химии и машинного обучения, что на практике нередко вызывает определённые трудности.

# СВЕДЕНИЯ О ДАННЫХ

***Описание предоставляемых данных:***

Химиками были предоставлены конфиденциальные данные о 1000 химических соединений с указанием их эффективности против вируса гриппа. Параметры, характеризующие эффективность, обозначаются как «IC50», «CC50» и «SI».

***Источник предоставления данных:***

Данные предоставлены образовательной платформой «Skillfactory» совместно c НИУЯ МИФИ.

***Дополнительные сведения от источника:***

Обратите внимание, что значение «SI» рассчитывается на основе параметров «IC50» и «CC50». Подробную информацию об этих показателях можно найти в открытых источниках для более глубокого понимания контекста задачи.

Все остальные представленные признаки являются числовыми характеристиками химических соединений.

# 1 РАЗВЕДОЧНЫЙ АНАЛИЗ ДАННЫХ (EDA)

# Исследование данных

Данные представлены в размере:

* количество строк – 1001;
* количество столбцов – 213.

Типы данных столбцов:

* float64 – 107;
* int64 106.

Количество пропусков: 36.

Ниже представлена Таблица 1 с пропусками.

*Таблица 1 — Количество пропусков в столбцах*

|  |  |
| --- | --- |
| **Название столбца** | **Количество пропусков** |
| 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 |

Количество пустых значений мало по сравнению с количеством строк в таблице. Принимается решение об удалении пустых строк.

# Работа с выбросами

Опираясь на условие задачи: «Представим следующую ситуацию: химиками были предоставлены конфиденциальные данные о 1000 химических соединений с указанием их эффективности против вируса гриппа. Параметры, характеризующие эффективность, обозначаются как IC50, CC50 и SI.»

Делаем вывод о том, что основополагающими для поиска выбросов являются столбцы ['IC50', 'CC50', 'SI'], так как они происходят от всех неупомянутых колонок таблицы.

Рассмотрим распределение данных у вышеупомянутых параметров на Рисунке 1.

# 

Рисунок 1 — Диаграмма попарного распределения данных

Судя по распределению данных, у нас явно есть выбросы в столбце «SI». А также судя по графикам распределения, наблюдается высокая корреляция между данными.

Исходя из информации на Рисунке 1, все 3 распределения напоминают логнормальное распределение с наличием потенциальных выбросов-«пеньков», далеко отстоящих от основной массы наблюдений.

На Рисунке 2 мы подобрали оптимальные значения сигмы и очистили данные от выбросов.

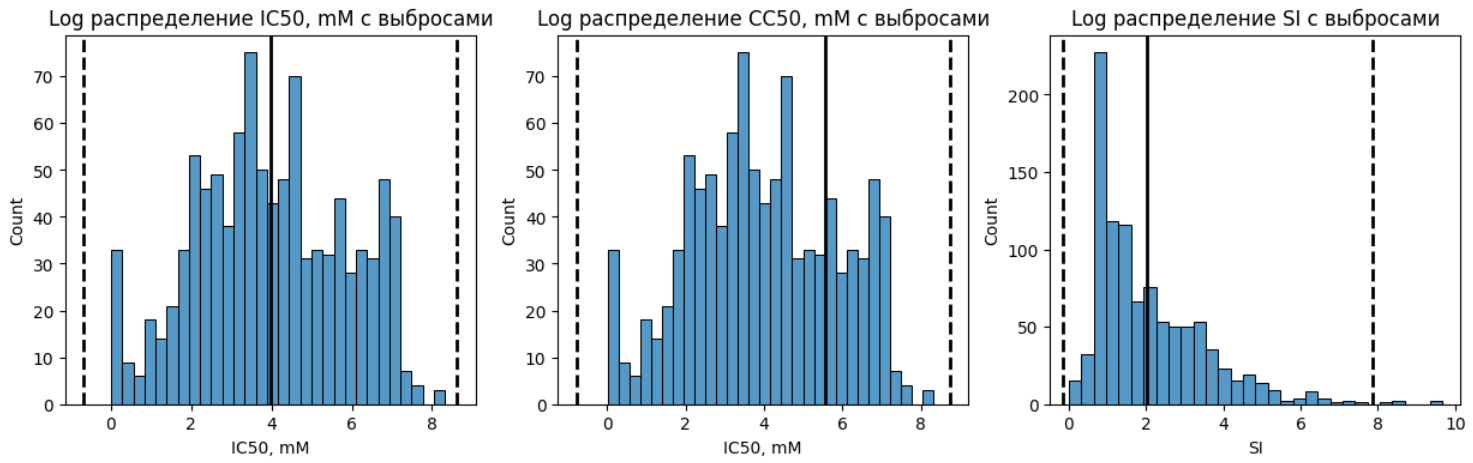


Рисунок 2 — Графики распределения с выбросами

## Feature Engineering

Для анализа данных мы составили тепловую карту, построенную на основе корреляционной матрицы параметров таблицы. Здесь красный — 100 % корреляция, синий — -100 % корреляция, то есть обратная, и серый — 0% слабая корреляция.

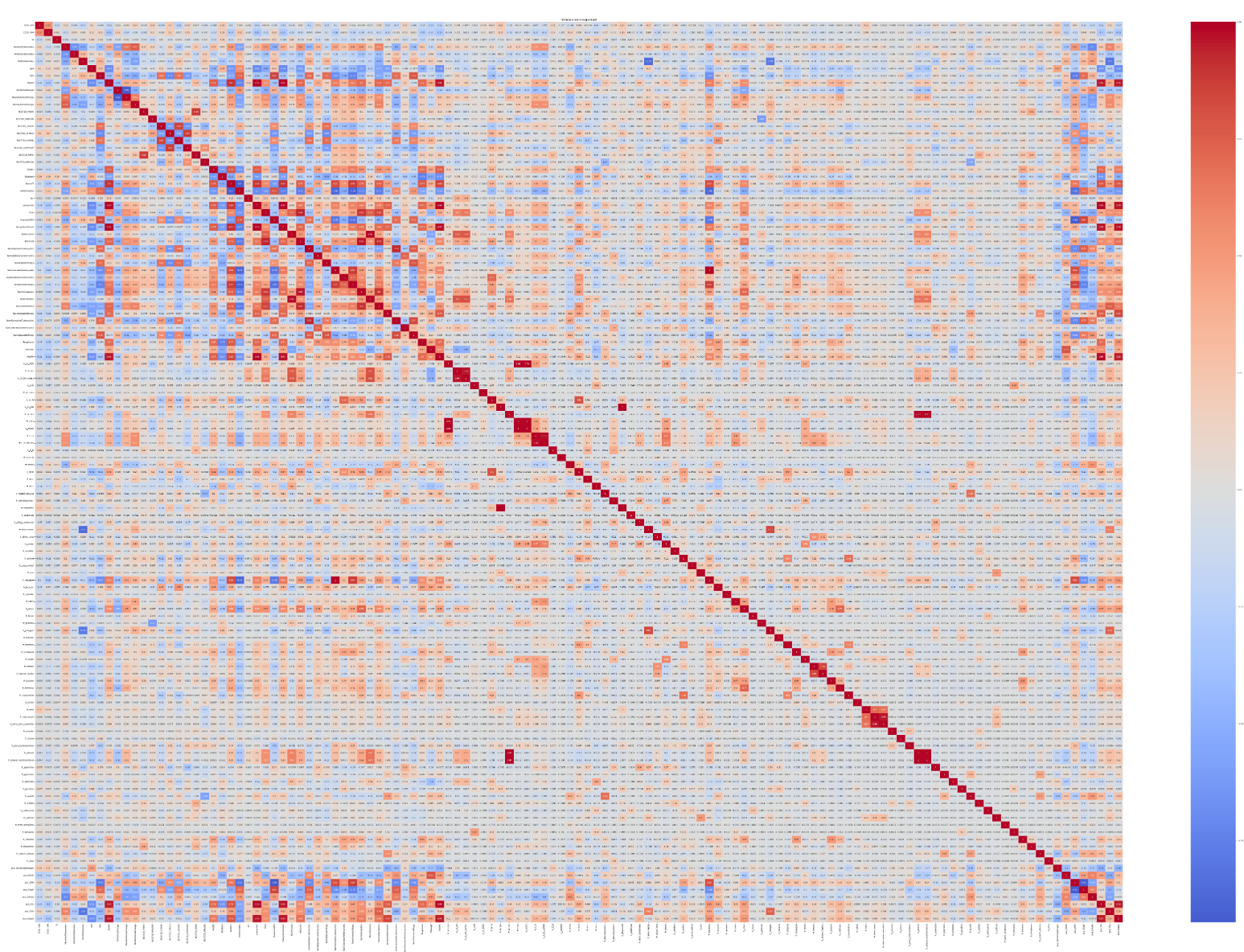


Рисунок 3 — Тепловая карта

После всех преобразований мы добились сокращения параметров с 213 до 123. Мы допускаем возможность дополнительного исследования по сокращению параметров и не утверждаем, что установленное количество столбцов является минимальным.

На текущий момент можно заявить, что уменьшение данных положительно отразиться на скорости обучения модели.

***Итоги по главе:***

Мы провели разведочный анализ данных, а значит готовы решать задачи регрессии и классификации. В ходе нашего анализа мы очистили данные, создали новые признаки и уменьшили мультиколлинеарность в данных.

# 2 ЗАДАЧИ РЕГРЕССИИ

Следующем этапом нашей работы является решение задачи регрессии. Целью решения текущей задачи является предсказывание целевой переменной при помощи алгоритмов машинного обучения, выбор лучшей модели, опираясь на статистические метрики, подбор наилучших входных значений для алгоритмов машинного обучения.

Целевые переменные для предсказания:

* CC50, mM;
* IC50, mM;
* SI.

Для каждой задачи вы разделили датасет на тестовый и тренировочный и применили к данным нормализацию, чтобы уменьшить шанс переобучения.

В рамках решения задачи регрессии мы используем одинаковые алгоритмы обучения и преобразования для каждой целевой переменной.

Качество предсказание мы оценивали при помощи mean average error (далее —MAE).

Алгоритмы регрессии, используемые для решения задачи:

* Linear Regression
* Ridge Regression
* Lasso Regression
* ElasticNet
* Decision Tree
* Random Forest
* Gradient Boosting

***Задача CC50:***

После обучения мы получили значения, представленные в Таблице 2.

*Таблица 2 — Результаты обучения CC50*

| **Алгоритм** | **MAE** |
| --- | --- |
| Linear Regression | 308.6705 |
| Ridge Regression | 284.6361 |
| Lasso Regression | 281.1711 |
| ElasticNet | 279.9608 |
| Decision Tree | 124.0530 |
| Random Forest | 81.7298 |
| Gradient Boosting | 62.2709 |

Лучше всех оказался «Gradient Boosting». В отличие от линейных моделей «Linear», «Ridge», «Lasso» градиентный бустинг не предполагает линейную связь между признаками и целевой переменной.

***Задача IC50:***

После обучения мы получили значения, представленные в Таблице 3.

*Таблица 3 — Результаты обучения IC50*

| **Алгоритм** | **MAE** |
| --- | --- |
| Linear Regression | 227.7363 |
| Ridge Regression | 210.5423 |
| Lasso Regression | 201.8229 |
| ElasticNet | 207.3508 |
| Decision Tree | 32.1039 |
| Random Forest | 16.4715 |
| Gradient Boosting | 16.4367 |

Небольшая разница между Random Forest и Gradient Boosting, это обусловлено тем, что скорее всего данные для IC50 лучше аппроксимируются из-за линейной зависимости.

Для определения какая из моделей является лучше мы используем кросс-валидацию. Этот тест покажет какая из моделей работает стабильнее. Результаты теста кросс-валидации представлены в Таблице 4.

*Таблица 4 — Результаты кросс-валидации*

| **Алгоритм** | **MAE (Cross-validation)** |
| --- | --- |
| Random Forest | 30.432942879723036 |
| Gradient Boosting | 26.872924653193838 |

Результат «Gradient Boosting» оказался более стабильным, «Gradient Boosting» предпочтителен для финальной версии.

***Задача SI:***

После обучения мы получили значения, представленные в Таблице 5.

*Таблица 5 — Результаты обучения SI*

| **Алгоритм** | **MAE** |
| --- | --- |
| Linear Regression | 43.3183 |
| Ridge Regression | 36.8357 |
| Lasso Regression | 36.1344 |
| ElasticNet | 37.2732 |
| Decision Tree | 20.7129 |
| Random Forest | 13.6843 |
| Gradient Boosting | 15.0458 |

Random Forest показал себя лучше всех, но и все остальные алгоритмы заметно лучше справились с задачей, связано это снова с линейностью в данных, так как из условия задачи мы знаем, что SI образуется из CC50 и IC50.

***Итоги по главе:***

Подводя итоги, мы можем сказать, что мы решили все поставленные задачи регрессии и выбрали лучшую модель на основе статистических показателей.

# 3 ЗАДАЧИ КЛАССИФИКАЦИИ

Следующем этапом нашей работы является решение задачи классификации. Целью решения текущей задачи является предсказывание бинарного класса целевой переменной при помощи алгоритмов машинного обучения, выбор лучшей модели, опираясь на статистические метрики, подбор наилучших входных значений для алгоритмов машинного обучения.

Целевые переменные для предсказания:

* CC50, mM > медианы;
* IC50, mM > медианы;
* SI > медианы;
* SI > 8.

Для каждой задачи вы разделили датасет на тестовый и тренировочный и применили к данным нормализацию, чтобы уменьшить шанс переобучения.

В рамках решения задачи регрессии мы используем одинаковые алгоритмы обучения и преобразования для каждой целевой переменной.

Качество предсказание мы оценивали при помощи Accuracy.

Алгоритмы регрессии, используемые для решения задачи:

* Logistic Regression;
* Ridge Classifier;
* Decision Tree;
* Random Forest;
* Gradient Boosting;
* SVM.

***Задача CC50 > медианы:***

После обучения мы получили значения, представленные в Таблице 6.

*Таблица 6 — Результаты обучения СС50 > медианы*

| **Алгоритм** | **Accuracy** |
| --- | --- |
| Logistic Regression | 0.82 |
| Ridge Classifier | 0.82 |
| Decision Tree | 0.98 |
| Random Forest | 0.85 |
| Gradient Boosting | 0.99 |
| SVM | 0.82 |

Gradient Boosting Accuracy — 0.99, возможно, мы переобучились, требуется проверить это через кросс-валидацию или на отдельной валидационной выборке.

После проверки с помощью крос-валидации Accuracy — 0.96. Может быть небольшое переобучение, но в целом модель работает очень хорошо.

***Задача IC50 > медианы:***

После обучения мы получили значения, представленные в Таблице 7.

*Таблица 7 — Результаты обучения IС50 > медианы*

| **Алгоритм** | **Accuracy** |
| --- | --- |
| Logistic Regression | 0.93 |
| Ridge Classifier | 0.76 |
| Decision Tree | 0.96 |
| Random Forest | 0.90 |
| Gradient Boosting | 0.97 |
| SVM | 0.76 |

Gradient Boosting Accuracy — 0.97, возможно, мы переобучились, требуется проверить это через кросс-валидацию или на отдельной валидационной выборке.

После проверки с помощью крос-валидации Accuracy — 0.9622. Может быть небольшое переобучение, но в целом модель работает очень хорошо.

***Задача SI > медианы:***

После обучения мы получили значения, представленные в Таблице 8.

*Таблица 8 — Результаты обучения SI > медианы*

| **Алгоритм** | **Accuracy** |
| --- | --- |
| Logistic Regression | 0.94 |
| Ridge Classifier | 0.77 |
| Decision Tree | 0.92 |
| Random Forest | 0.81 |
| Gradient Boosting | 0.94 |
| SVM | 0.83 |

Gradient Boosting и Logistic Regression — лучший результат. Gradient Boosting Accuracy — 0.9430, Logistic Regression Accuracy — 0.9430. Проверим стабильность обоих моделей с помощью кросс-валидации

Результат Gradient Boosting CV Accuracy: 0.9492148374501316 оказался более стабильным, Gradient Boosting предпочтителен для финальной версии

***Задача SI > 8:***

После обучения мы получили значения, представленные в Таблице 9.

*Таблица 9 — Результаты обучения SI > 8*

| **Алгоритм** | **Accuracy** |
| --- | --- |
| Logistic Regression | 0.96 |
| Ridge Classifier | 0.80 |
| Decision Tree | 0.96 |
| Random Forest | 0.83 |
| Gradient Boosting | 0.97 |
| SVM | 0.82 |

Gradient Boosting и Logistic Regression — очень близкие результаты. Gradient Boosting Accuracy = 0.9689, Logistic Regression Accuracy = 0.9637. Модели почти без ошибок определяют оба класса. Проверим стабильность обоих моделей с помощью кросс-валидации.

Результат Gradient Boosting CV Accuracy: 0.9518207282913165 оказался более стабильным чем Logistic Regression CV Accuracy: 0.9297173414820474, Gradient Boosting предпочтителен для финальной версии.

***Итоги по главе:***

Подводя итоги, мы можем сказать, что мы решили все поставленные задачи классификации и выбрали лучшую модель на основе статистических показателей.

# 4 ЗАКЛЮЧЕНИЕ

Подводя итоги работы, мы можем сказать, что провели разведочный анализ данных, решили задачи классификации и регрессии. В ходе нашего анализа мы очистили данные, создали новые признаки и уменьшили мультиколлинеарность в данных.

В ходе решения задач регрессии и классификации в большинстве задач лидировала модель «Gradient Boosting», как мы уже писали ранее, это связано с особенностью алгоритма решать задачи с нелинейной зависимостью в данных.

При решении задач использовались такие метрики как: MAE в задачах регрессии, Accuracy в задачах классификации.

Все поставленные задачи выполнены в соответствии с условием.