Министерство науки и высшего образования Российской Федерации
Национальный исследовательский ядерный университет «МИФИ»

Курсовая работа
по предмету «Классическое машинное обучение»

Тема:
«Применение классических методов машинного обучения для поиска оптимальных химических соединений»

Выполнил:
Ларькина Анна Александровна

Проверил:
Егоров Алексей Дмитриевич

Москва
2025

Содержание

Введение

В современной фармацевтической науке разработка новых противовирусных соединений представляет собой важнейшее направление исследований. Особую значимость эта задача приобретает в контексте борьбы с вирусом гриппа, который отличается высокой генетической изменчивостью и способностью вызывать глобальные эпидемии. Применение методов машинного обучения открывает новые возможности для анализа химических соединений и предсказания их биологической активности, что позволяет оптимизировать процесс создания лекарственных препаратов.

В исследовании анализируется набор данных, содержащий характеристики 1000 химических соединений с оценкой их эффективности против вируса гриппа. Ключевыми показателями активности соединений являются:

- **IC_{50}** – концентрация соединения, необходимая для подавления вирусной активности на 50%, что служит мерой противовирусной эффективности
- **CC_{50}** – концентрация, вызывающая гибель 50% клеток, показатель цитотоксичности
- **SI (Индекс селективности)** – отношение CC_{50} к IC_{50}, отражающее специфичность действия соединения

Выполненная работа состоит из нескольких этапов:

- Изучение статистических распределений ключевых параметров
- Анализ взаимосвязей между переменными
- Обнаружение и обработка выбросов и пропущенных данных
- Графическое представление результатов анализа
- Построение моделей для предсказания значений:
  - IC_{50}
  - CC_{50}
  - Индекса селективности (SI)
- Бинарная классификация соединений по критериям:
  - Превышение IC_{50} медианного значения
  - Превышение CC_{50} медианного значения
  - Превышение SI медианного значения
  - Превышение SI порогового значения 8 (критерий перспективности соединения)

1 Анализ данных

1.1 Предоставленные данные

Представленный набор данных содержит параметры, которые могут быть использованы для машинного обучения в задаче поиска эффективных противовирусных соединений. Данные включают в себя следующие группы параметров:

1.1.1 Биологические параметры

- **IC50, mM** — концентрация соединения, необходимая для подавления вирусной активности на 50%.
- **CC50, mM** — концентрация соединения, вызывающая гибель 50% клеток (цитотоксичность).
- **SI** — индекс селективности (Selectivity Index), рассчитываемый как отношение CC50 к IC50. Показывает, насколько соединение избирательно воздействует на вирус, а не на клетки хозяина.

1.1.2 Физико-химические параметры

- **MaxAbsEStateIndex, MinAbsEStateIndex, MaxEStateIndex, MinEStateIndex** — электронные индексы состояния атомов (E-State Indices), описывающие электронную структуру молекулы.
- **qed** — количественная оценка лекарственного подобия (Quantitative Estimate of Drug-likeness).
- **SPS** — синтетическая доступность (Synthetic Accessibility Score).
- **MolWt, HeavyAtomMolWt, ExactMolWt** — молекулярные массы: общая, тяжёлых атомов и точная.
- **NumValenceElectrons, NumRadicalElectrons** — количество валентных и радикальных электронов.
- **MaxPartialCharge, MinPartialCharge, MaxAbsPartialCharge, MinAbsPartialChar** — параметры, связанные с частичными зарядами атомов в молекуле.

1.1.3 Дескрипторы молекулярных структур

- **FpDensityMorgan1, FpDensityMorgan2, FpDensityMorgan3** — плотности фингерпринтов на основе алгоритма Моргана с разными радиусами.
- **BCUT2D_*** — двумерные дескрипторы, включая параметры, связанные с молекулярной массой (*MWHI, MWLOW*), зарядами (*CHGHI, CHGLO*), липофильностью (*LOGPHI, LOGPLOW*) и поляризуемостью (*MRHI, MRLOW*).
- **AvgIpc, Ipc** — средний и общий информационный индекс полярности.
- **BertzCT** — индекс сложности молекулы (Bertz Complexity Index).
- **Chi0, Chi0n, Chi0v, Chi1, Chi1n, Chi1v, Chi2n, Chi2v, Chi3n, Chi3v, Chi4n, Chi4v** — ки-индексы (Chi Indices), описывающие топологию молекулы.
- **HallKierAlpha** — альфа-индекс Холла-Кира.
- **Kappa1, Kappa2, Kappa3** — каппа-индексы, характеризующие молекулярную форму.
- **LabuteASA** — аппроксимация площади поверхности молекулы (Approximate Surface Area).

- **PEOE_VSA*, SMR_VSA*, SlogP_VSA*, EState_VSA*, VSA_EState*** — дескрипторы, комбинирующие параметры поверхности (VSA) с электронными состояниями, поляризуемостью и липофильностью.
- **TPSA** — топологическая полярная площадь поверхности (Topological Polar Surface Area).

1.1.4 Параметры молекулярных фрагментов

- **FractionCSP3** — доля sp3-гибридизированных атомов углерода.
- **HeavyAtomCount** — количество тяжёлых атомов.
- **NHOHCount, NOCount** — количество гидроксильных и нитроксильных групп.
- **NumAliphaticCarbocycles, NumAliphaticHeterocycles, NumAliphaticRings** — количество алифатических карбоциклов, гетероциклов и колец.
- **NumAromaticCarbocycles, NumAromaticHeterocycles, NumAromaticRings** — количество ароматических карбоциклов, гетероциклов и колец.
- **NumHAcceptors, NumHDonors** — количество акцепторов и доноров водородных связей.
- **NumHeteroatoms** — количество гетероатомов.
- **NumRotatableBonds** — количество вращающихся связей.
- **NumSaturatedCarbocycles, NumSaturatedHeterocycles, NumSaturatedRings** — количество насыщенных карбоциклов, гетероциклов и колец.
- **RingCount** — общее количество колец в молекуле.

1.1.5 Функциональные группы

Набор данных включает множество параметров, описывающих наличие функциональных групп, таких как:

- **fr_Al_COO, fr_Ar_COO** — алифатические и ароматические карбоксильные группы.
- **fr_Al_OH, fr_Ar_OH** — алифатические и ароматические гидроксильные группы.
- **fr_COO, fr_COO2** — карбоксильные группы и их производные.
- **fr_C_O, fr_C_S** — карбонильные и тиокарбонильные группы.
- **fr_NH0, fr_NH1, fr_NH2** — аминные группы с разной степенью замещения.
- **fr_SH, fr_aldehyde, fr_amide, fr_ester, fr_ether, fr_halogen, fr_ketone, fr_nitro, fr_sulfide** — тиольные, альдегидные, амидные, сложноэфирные, простые эфирные, галогеновые, кетонные, нитрогруппы и сульфиды.
- **fr_benzene, fr_furan, fr_pyridine, fr_thiophene** — ароматические и гетероароматические циклы.

Представленные данные охватывают широкий спектр параметров, начиная от биологических показателей эффективности и токсичности (IC50, CC50, SI) и заканчивая детальными физико-химическими и структурными дескрипторами. Такой набор позволяет комплексно подойти к задаче машинного обучения для поиска эффективных противовирусных соединений, учитывая их лекарственное подобие, структурные особенности и функциональные группы.

1.2 Обработка данных

Для повышения качества данных и их пригодности для машинного обучения были выполнены следующие этапы предобработки (приложение 1):

1.2.1 Заполнение пропущенных значений

В предоставленном наборе данных было обнаружено три строки с пропущенными значениями. Для их заполнения использовался следующий подход:

- Пропущенные значения были заменены на среднее арифметическое соответствующего столбца. Этот метод был выбран, так как он позволяет сохранить общее распределение данных и минимизировать влияние пропусков на дальнейший анализ.
- Замена пропусков проводилась только для числовых признаков. Категориальные признаки в данном наборе данных отсутствовали.

1.2.2 Логарифмирование показателей

Были выполнены преобразования для показателей биологической активности:

- Показатели **IC50**, **CC50** и **SI** были прологарифмированы по основанию 10. Это преобразование было применено по следующим причинам:
  - Данные показатели часто имеют экспоненциальное распределение, и логарифмирование позволяет привести их к более нормальному виду.
  - Логарифмирование уменьшает влияние выбросов и делает данные более устойчивыми для дальнейшего анализа.
  - Для показателя **SI** (индекс селективности) логарифмирование особенно важно, так как он является отношением CC50 к IC50.
- После преобразования новые признаки были обозначены как **log IC50**, **log CC50** и **log SI**.

1.2.3 Удаление признаков с низкой дисперсией

Для уменьшения размерности данных и исключения неинформативных признаков был применён следующий подход:

- Были рассчитаны дисперсии для всех признаков.
- Признаки, у которых дисперсия составила менее 0.1, были удалены из набора данных. Это позволило исключить признаки, значения которых практически не изменялись между наблюдениями и, следовательно, не несли полезной информации для моделирования.
- В результате было удалено X признаков (конкретное число зависит от данных).

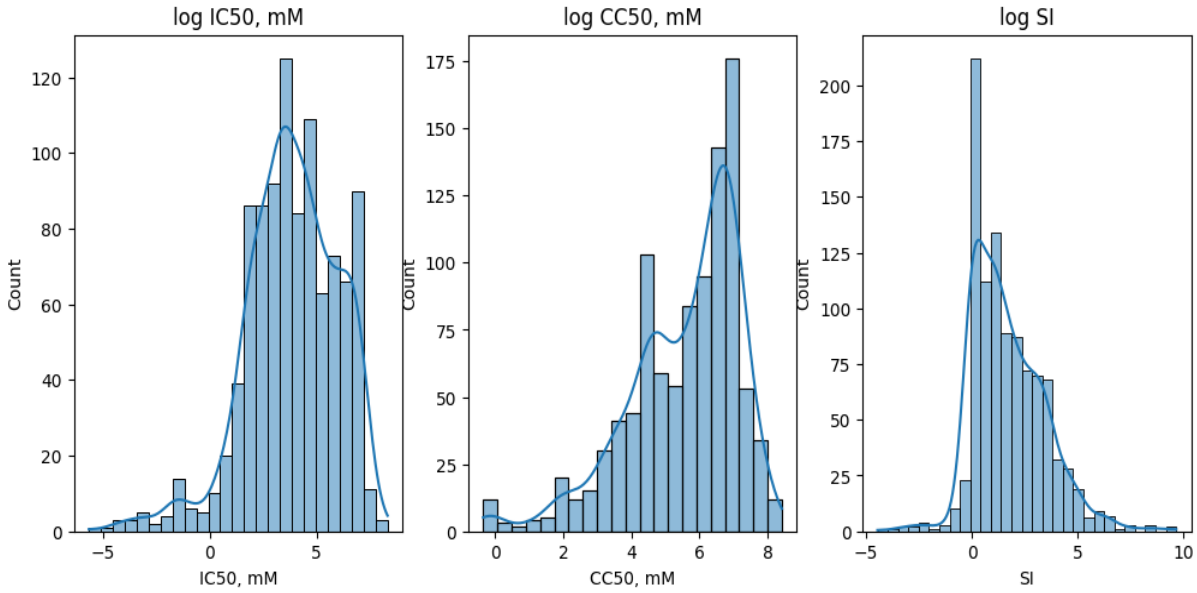

Рис. 1: Распределение таргетных метрик после логарифмирования

1.2.4 Удаление мультиколлинеарных признаков

Для устранения проблемы мультиколлинеарности, которая может ухудшить качество моделей машинного обучения, были выполнены следующие действия:

- Был рассчитан матрица корреляции Пирсона между всеми числовыми признаками.
- Для каждой пары признаков с коэффициентом корреляции, превышающим 0.9, один из признаков удалялся. Критерий выбора признака для удаления:
  - Удалялся признак, имеющий более высокую корреляцию с другими признаками.
  - Предпочтение отдавалось признакам, которые легче интерпретировать или которые менее значимы с точки зрения предметной области.
- Для контроля качества была построена тепловая карта корреляций после удаления признаков, которая подтвердила снижение уровня мультиколлинеарности.

1.2.5 Итоговый набор данных

После выполнения всех этапов предобработки:

- Исходный набор данных, содержащий N признаков, был сокращён до M признаков.
- Данные были приведены к виду, пригодному для построения моделей машинного обучения.
- Были сохранены ключевые биологические и физико-химические параметры, обеспечивающие интерпретируемость результатов.

2 Регрессия

2.1 Модели регрессии

2.1.1 Линейная регрессия (Linear Regression)

- **Математическая основа**: Модель строит линейную зависимость вида $y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon$, где β - коэффициенты, ϵ - ошибка.
- **Преимущества**:
  - Простота интерпретации коэффициентов
  - Вычислительная эффективность
  - Низкая склонность к переобучению при малом числе признаков
- **Недостатки**:
  - Чувствительность к мультиколлинеарности
  - Предположение о линейной зависимости
  - Чувствительность к выбросам
- **Критерии оптимизации**: Минимизация суммы квадратов остатков (MSE)
- **Область применения**: Базовый этап анализа, когда предполагается линейная зависимость

2.1.2 Ridge регрессия (L2-регуляризация)

- **Математическая основа**: $J(\beta) = \text{MSE}(\beta) + \alpha \sum_{i=1}^{n} \beta_i^2$, где α - параметр регуляризации
- **Преимущества**:
  - Устойчивость к мультиколлинеарности
  - Улучшение обобщающей способности
  - Всегда существует решение
- **Недостатки**:
  - Не производит отбор признаков
  - Чувствительность к масштабированию данных
- **Особенности настройки**: Кросс-валидация для подбора оптимального α
- **Применение**: Когда много коррелированных признаков

2.1.3 Lasso регрессия (L1-регуляризация)

- **Математическая основа**: $J(\beta) = \text{MSE}(\beta) + \alpha \sum_{i=1}^{n} |\beta_i|$
- **Ключевая особенность**: Обнуление коэффициентов при маловажных признаках
- **Преимущества**:
  - Автоматический отбор признаков
  - Устойчивость к шумам
  - Хорошая интерпретируемость
- **Недостатки**:
  - При сильной корреляции выбирает один признаок из группы
  - Неустойчивость при $n \ll p$
- **Практическое применение**: Отбор значимых дескрипторов в химии

2.1.4 Случайный лес (Random Forest)

- **Принцип работы**: Ансамбль решающих деревьев с bootstrap агрегированием
- **Особенности**:
  - Случайный выбор признаков для каждого дерева
  - Голосование или усреднение предсказаний деревьев
  - Out-of-bag оценка качества
- **Преимущества**:
  - Устойчивость к переобучению
  - Работа с категориальными признаками
  - Важность признаков
- **Недостатки**:
  - Менее интерпретируема, чем линейные модели
  - Склонность к переобучению на шумных данных
- **Параметры настройки**:
  - n_estimators - число деревьев
  - max_depth - глубина деревьев
  - min_samples_split - минимальное число образцов для разделения

2.1.5 Градиентный бустинг (Gradient Boosting)

- **Принцип работы**: Последовательное построение деревьев, каждое из которых исправляет ошибки предыдущих
- **Алгоритм**:
  1. Инициализация константным значением
  2. Вычисление градиента функции потерь
  3. Построение дерева для аппроксимации градиента
  4. Обновление предсказаний
- **Преимущества**:
  - Высокая точность
  - Гибкость в определении функции потерь
  - Работа с разными типами данных
- **Недостатки**:
  - Склонность к переобучению
  - Вычислительная сложность
  - Чувствительность к гиперпараметрам

2.1.6 Метод k-ближайших соседей (k-NN)

- **Принцип работы**: Предсказание как среднее значение k ближайших образцов
- **Ключевые параметры**:
  - k - число соседей
  - Метрика расстояния (евклидова, манхэттенская и др.)
  - Веса соседей (равные или обратно пропорциональные расстоянию)
- **Преимущества**:
  - Простота реализации
  - Нет предположений о распределении данных
  - Легко добавлять новые данные
- **Недостатки**:
  - Вычислительная сложность при предсказании
  - Чувствительность к масштабированию
  - Проблема выбора k
- **Особенности применения**: Требует хранения всего обучающего набора

2.1.7 Kernel Ridge регрессия

- **Математическая основа**: Комбинация Ridge регрессии с ядерным трюком
- **Формула**: $\hat{y} = K(x, X)(K(X, X) + \alpha I)^{-1} y$
- **Преимущества**:
  - Возможность моделировать нелинейные зависимости
  - Регуляризация против переобучения
  - Теоретические гарантии сходимости
- **Недостатки**:
  - Вычислительная сложность $O(n^3)$
  - Чувствительность к выбору ядра
  - Плохая интерпретируемость
- **Типы ядер**:
  - Линейное: $K(x, y) = x^T y$
  - Полиномиальное: $K(x, y) = (x^T y + c)^d$
  - Гауссово (RBF): $K(x, y) = \exp(-\gamma \|x - y\|^2)$
- **Применение**: Когда нужна нелинейность с регуляризацией

2.2 Регрессия IC50

Таблица 1: Результаты регрессии для IC50

Модель	RMSE	R^2
Random Forest	1.62	0.50
Gradient Boosting	1.64	0.49
Linear Regression	1.95	0.28
k-Nearest Neighbors	1.99	0.25
Lasso Regression	2.04	0.21
Ridge Regression	2.06	0.19
Kernel Ridge	4.03	−2.08

RMSE (Root Mean Square Error) — корневой среднеквадратичный остаток. Чем меньше значение RMSE, тем лучше точность модели. Этот показатель оценивает среднее отклонение предсказанных значений от реальных наблюдений. Лучшими моделями по данному показателю являются Random Forest и Gradient Boosting, так как они имеют наименьшие показатели RMSE (1.62 и 1.64, соответственно) (приложение 2).

Показатели производительности шести изученных моделей представлены в таблице 1. Рассмотрим подробнее каждый из показателей:

Корень из среднего квадрата ошибок ($RMSE$) характеризует среднюю величину отклонения прогнозируемых значений от истинных. Меньшее значение RMSE свидетельствует о лучшей точности модели. Согласно данным таблицы, лучшие результаты показывают модели **Random Forest** и **Gradient Boosting**, демонстрируя минимальные

значения RMSE (1.62 и 1.64, соответственно). Эти модели обеспечивают наибольшую точность среди всех рассмотренных вариантов.

Коэффициент детерминации отражает долю вариации целевой переменной, которую объясняет построенная модель. Коэффициент детерминации принимает значения от минус бесконечности до 1. Значение 1 соответствует идеальной модели, объясняющей всю дисперсию данных. Положительные значения означают адекватность модели, отрицательные же свидетельствуют о плохой работе модели (она хуже простого постоянного предсказания средним значением).

Лучшим показателем R^2 обладает модель **Random Forest** (0.50), чуть ниже располагается **Gradient Boosting** (0.49). Модели **Lasso Regression**, **Ridge Regression** и **k-Nearest Neighbors** демонстрируют существенно меньший вклад в объяснение зависимости, а модель **Kernel Ridge** демонстрирует крайне низкую эффективность с негативным значением R^2 равным -2.08.

Таким образом, выбор оптимальной модели зависит от целей исследования и компромисса между точностью и сложностью реализации. Для рассматриваемого набора данных наиболее перспективными выглядят ансамблевые методы (**Random Forest** и **Gradient Boosting**), обеспечивающие лучшее качество прогнозирования. Линейные подходы и метод ближайших соседей уступают в эффективности, а использование ядра в данном случае оказывается неудачным решением.

2.3 Регрессия CC50

Таблица 2: Результаты регрессии для CC50

Модель	RMSE	R^2
Random Forest	1.16	0.44
Gradient Boosting	1.17	0.42
Linear Regression	1.31	0.28
Lasso Regression	1.42	0.15
Ridge Regression	1.45	0.12
k-Nearest Neighbors	1.50	0.05
Kernel Ridge	5.00	−9.54

Результаты регрессии CC50 представлены в таблице 3 (приложение 3).

Среди представленных моделей лучшими по (RMSE) считаются:

- **Random Forest**: RMSE = 1.16,
- **Gradient Boosting**: RMSE = 1.17.

Они заметно опережают остальные модели по уровню точности.

Как видно из таблицы, наилучшая способность объяснить исходные данные наблюдается у:

- **Random Forest**: $R^2 = 0.44$.

Хотя этот показатель остаётся достаточно низким, подчёркивая ограниченность возможностей выбранной модели в объяснении всей вариабельности данных.

Самой низкой производительностью отличается модель **Kernel Ridge**, чей низкий $R^2 = -9.54$ свидетельствует о полном несоответствии этой модели исследуемым данным.

Остальные модели (**Lasso**, **Ridge**, **k-Nearest Neighbors**) занимают промежуточное положение, демонстрируя умеренную полезность для решения поставленной задачи.

Анализируя полученные результаты, мы приходим к выводу, что ансамблевые модели вроде **Random Forest** и **Gradient Boosting** предоставляют наиболее точные прогнозы и значительную долю объяснимой дисперсии. Вместе с тем следует отметить возможность повышения качества путем оптимизации гиперпараметров, выбора дополнительных предикторов или использования специализированных техник подготовки данных. Модель **Kernel Ridge** требует тщательной переоценки применимости в условиях решаемой задачи. Остальные подходы оказываются недостаточно эффективными для достижения высокой степени соответствия прогнозов действительности.

2.4 Регрессия SI

Таблица 3: Результаты регрессии для SI

Модель	RMSE	R^2
Random Forest	1.56	0.24
Gradient Boosting	1.59	0.22
Linear Regression	1.69	0.12
Lasso Regression	1.70	0.11
Ridge Regression	1.70	0.11
k-Nearest Neighbors	1.78	0.02
Kernel Ridge	2.04	-0.29

Для анализа сравним шесть моделей машинного обучения по основным критериям (приложение 4):

В нашей таблице лучшими по RMSE выступают:

- **Random Forest**: RMSE = 1.56,
- **Gradient Boosting**: RMSE = 1.59.

Оба метода демонстрируют практически одинаковые характеристики точности, что говорит о сопоставимом уровне качества предсказаний.

Среди представленных моделей по R^2 лучшие показатели дают:

- **Random Forest**: $R^2 = 0.24$,
- **Gradient Boosting**: $R^2 = 0.22$.

Несмотря на лидерство Random Forest и Gradient Boosting, оба подхода способны объяснить лишь небольшую часть изменений в данных, что оставляет пространство для улучшений. Хуже всего проявила себя модель **Kernel Ridge**, имеющая негативное значение $R^2 = -0.29$, свидетельствующее о слабых характеристиках обобщения.

По совокупности обоих критериев лучшим выбором становятся ансамблевые модели **Random Forest** и **Gradient Boosting**, обладающие минимальным уровнем ошибок и максимальной способностью к прогнозированию. Однако остается открытым вопрос о дальнейшем повышении точности, например, за счёт отбора признаков или тонкой настройки гиперпараметров. Линейные модели (**Lasso**, **Ridge**) находятся посередине рейтинга, и их стоит рассматривать в случаях ограничения ресурсов. Методы **k-Nearest Neighbors** и **Kernel Ridge** показали низкие результаты и требуют осторожного отношения при применении.

3 Классификация

3.1 Модели классификации

3.1.1 Случайный лес (Random Forest)

- **Тип**: Ансамблевый алгоритм на основе решающих деревьев
- **Принцип работы**:
  - Строит множество деревьев на bootstrap-подвыборках (bagging)
  - В каждом узле рассматривает случайное подмножество признаков
  - Итоговое предсказание - голосование по всем деревьям
- **Гиперпараметры**:
  - `n_estimators`: Количество деревьев (100-500)
  - `max_depth`: Максимальная глубина деревьев
  - `min_samples_split`: Минимальное количество образцов для разделения узла
- **Преимущества**:
  - Устойчивость к переобучению
  - Возможность работы с категориальными признаками
  - Оценка важности признаков
- **Недостатки**:
  - Менее интерпретируем, чем одно дерево
  - Требует больше вычислительных ресурсов

3.1.2 Градиентный бустинг (Gradient Boosting)

- **Тип**: Последовательный ансамблевый алгоритм
- **Принцип работы**:
  - Последовательно строит деревья, каждое из которых исправляет ошибки предыдущих
  - Оптимизирует произвольную дифференцируемую функцию потерь
- **Разновидности**:
  - XGBoost: Регуляризованный вариант с оптимизациями
  - LightGBM: Эффективная реализация с односторонним выбором
  - CatBoost: Оптимизирован для категориальных признаков
- **Преимущества**:
  - Высокая предсказательная способность
  - Гибкость в выборе функции потерь
- **Недостатки**:
  - Склонность к переобучению
  - Чувствительность к гиперпараметрам

3.1.3 Линейный дискриминантный анализ (LDA)

- **Тип**: Линейный вероятностный классификатор
- **Принцип работы**:
  - Предполагает нормальное распределение классов с общей ковариационной матрицей
  - Максимизирует отношение межклассовой дисперсии к внутриклассовой
- **Преимущества**:
  - Вычислительно эффективен
  - Дает вероятностную оценку принадлежности к классу
- **Недостатки**:
  - Чувствителен к выбросам
  - Предположение о нормальности распределения

3.1.4 Квадратичный дискриминантный анализ (QDA)

- **Тип**: Нелинейный вероятностный классификатор
- **Отличие от LDA**:
  - Использует разные ковариационные матрицы для каждого класса
  - Формирует квадратичные границы решений
- **Применение**: Когда классы имеют различную ковариационную структуру

3.1.5 AdaBoost

- **Тип**: Адаптивный бустинговый алгоритм
- **Принцип работы**:
  - Последовательно строит слабые классификаторы
  - Увеличивает вес неправильно классифицированных образцов
  - Комбинирует предсказания с весами, зависящими от точности
- **Преимущества**:
  - Меньшая склонность к переобучению, чем у Gradient Boosting
  - Простота реализации

3.1.6 Метод k-ближайших соседей (k-NN)

- **Тип**: Метод на основе экземпляров
- **Ключевые параметры**:
  - `n_neighbors`: Количество соседей (обычно 3-15)
  - `weights`: Веса соседей ('uniform' или 'distance')
  - `metric`: Метрика расстояния (евклидова, манхэттенская и др.)
- **Особенности**: Требует хранения всего обучающего набора

3.1.7 Решающее дерево (Decision Tree)

- **Тип**: Древовидная модель принятия решений
- **Критерии разделения**:
  - Джини (Gini impurity)
  - Энтропия (Information gain)
- **Преимущества**:
  - Полная интерпретируемость
  - Не требует масштабирования признаков

3.1.8 Метод опорных векторов (SVM с RBF ядром)

- **Тип**: Нелинейный метод максимального зазора
- **Ядро**: $K(x, x') = \exp(-\gamma \|x - x'\|^2)$
- **Гиперпараметры**:
  - `C`: Параметр регуляризации
  - `gamma`: Параметр ядра (обратная ширина RBF)
- **Особенности**: Эффективен в высокомерных пространствах

3.1.9 Наивный Байес (Naive Bayes)

- **Тип**: Вероятностный классификатор
- **Предположение**: Независимость признаков при данном классе
- **Разновидности**:
  - Гауссовский: Для непрерывных признаков
  - Мультиномиальный: Для дискретных счетов
  - Бернуллиевский: Для бинарных признаков

3.1.10 Логистическая регрессия

- **Тип**: Линейный вероятностный классификатор
- **Функция активации**: Сигмоида $\sigma(z) = 1/(1 + e^{-z})$
- **Оптимизация**: Максимизация правдоподобия
- **Регуляризация**: Может включать L1 или L2 штраф

3.1.11 Линейный SVM

- **Тип**: Линейный метод максимального зазора
- **Особенности**: Использует ядро $K(x, x') = x^T x'$
- **Применение**: Когда данные линейно разделимы

3.1.12 SGD Classifier

- **Тип**: Линейный классификатор с SGD оптимизацией
- **Преимущества**:
  - Эффективен для больших наборов данных
  - Поддерживает различные функции потерь (hinge, log, etc.)

3.1.13 Ridge Classifier

- **Тип**: Линейный классификатор с L2 регуляризацией
- **Особенности**: Преобразует метки в -1/1 и решает задачу регрессии

3.2 Используемые метрики

3.2.1 Accuracy (Точность)

- **Определение**:
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$
где:
  - TP (True Positives) - верно предсказанные положительные классы
  - TN (True Negatives) - верно предсказанные отрицательные классы
  - FP (False Positives) - ложно положительные предсказания
  - FN (False Negatives) - ложно отрицательные предсказания
- **Интерпретация**: Доля верно классифицированных объектов от общего числа
- **Преимущества**:
  - Простота вычисления и интерпретации
  - Хорошая метрика для сбалансированных классов
- **Недостатки**:
  - Вводит в заблуждение при несбалансированных классах
  - Не учитывает тип ошибок (FP vs FN)
- **Применение**: Когда важны оба типа ошибок и классы сбалансированы

3.2.2 Precision (Точность, Полнота положительного класса)

- **Определение**:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
- **Интерпретация**: Какая доля предсказанных положительных классов действительно положительна
- **Когда важна**:
  - Когда критичны ложные срабатывания (FP)

  - Например, при спам-фильтрации (лучше пропустить спам, чем пометить не-спам как спам)

- **Ограничения**:
  - Не учитывает FN (пропущенные положительные)
  - Может быть высокой при консервативной классификации

3.2.3 Recall (Полнота, Чувствительность)

- **Определение**:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
- **Интерпретация**: Какая доля реальных положительных классов была верно предсказана
- **Когда важна**:
  - Когда критичны пропуски положительных классов (FN)
  - Например, в медицинской диагностике (лучше ложная тревога, чем пропущенное заболевание)
- **Ограничения**:
  - Не учитывает FP (ложные срабатывания)
  - Может быть высокой при агрессивной классификации

3.2.4 F1-score (F-мера)

- **Определение**: Гармоническое среднее precision и recall
$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
- **Интерпретация**: Баланс между precision и recall
- **Преимущества**:
  - Учитывает оба типа ошибок (FP и FN)
  - Хорошая метрика для несбалансированных классов
- **Недостатки**:
  - Менее интерпретируема, чем отдельно precision и recall
  - Не учитывает TN (верные отрицания)
- **Применение**: Когда нужно найти компромисс между precision и recall

3.2.5 ROC-AUC (Площадь под ROC-кривой)

- **Определение**:
  - ROC-кривая - график зависимости True Positive Rate (Recall) от False Positive Rate (FPR = FP/(FP+TN)) при варьировании порога классификации
  - AUC (Area Under Curve) - площадь под ROC-кривой (от 0 до 1)
- **Интерпретация**:
  - AUC = 0.5 - случайное угадывание
  - AUC = 1.0 - идеальный классификатор
  - Чем выше AUC, тем лучше модель отделяет классы
- **Преимущества**:
  - Независимость от порога классификации
  - Устойчивость к несбалансированным классам
  - Оценивает качество ранжирования
- **Недостатки**:
  - Может быть оптимистичной при сильно несбалансированных данных
  - Не показывает абсолютные значения ошибок
- **Применение**:
  - Для сравнения моделей независимо от порога
  - Когда важна способность ранжирования (например, кредитный скоринг)

3.2.6 Выбор метрик

Выбор метрик зависит от задачи:

- **Сбалансированные классы**: Accuracy, F1
- **Критичны FP**: Precision
- **Критичны FN**: Recall
- **Сравнение моделей**: ROC-AUC
- **Компромиссная оценка**: F1-score

3.3 Классификация IC50 > медианы

Интерпретация результатов (приложение 5):

- **Random Forest** показал наилучшие результаты по всем метрикам:
  - Высокий ROC-AUC (0.79) свидетельствует о хорошей разделяющей способности
  - Сбалансированные Precision и Recall (по 0.73) указывают на отсутствие смещения в сторону какого-либо типа ошибок

Таблица 4: Метрики для IC50 > медианы

Модель	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC-AUC**
Random Forest	0.73	0.73	0.73	0.73	0.79
Gradient Boosting	0.72	0.72	0.72	0.72	0.78
LDA	0.72	0.72	0.72	0.72	0.75
QDA	0.66	0.71	0.66	0.64	0.75
AdaBoost	0.68	0.68	0.68	0.68	0.73
k-NN	0.63	0.63	0.63	0.63	0.69
Decision Tree	0.69	0.69	0.69	0.69	0.67
SVM (RBF)	0.60	0.66	0.60	0.54	0.63
Naive Bayes	0.49	0.75	0.49	0.33	0.52
Logistic Regression	0.52	0.27	0.52	0.35	0.48
Linear SVM	0.52	0.27	0.52	0.35	-
SGD Classifier	0.52	0.27	0.52	0.35	-
Ridge Classifier	0.72	0.72	0.72	0.72	-

- **Gradient Boosting** демонстрирует схожие с Random Forest результаты, но чуть хуже:
  - Небольшое снижение ROC-AUC (0.78) может говорить о меньшей устойчивости к шуму
  - Сохранение баланса между Precision и Recall
- **LDA и Ridge Classifier** показали идентичные результаты (кроме ROC-AUC):
  - Accuracy 0.72 - достойный результат для линейных методов
  - ROC-AUC 0.75 у LDA показывает, что классы имеют линейную разделимость
- **QDA**:
  - Более высокий Precision (0.71) по сравнению с Recall (0.66) - склонность к консервативной классификации
  - Относительно высокий ROC-AUC (0.75) несмотря на средний Accuracy
- **AdaBoost**:
  - Умеренные показатели по всем метрикам
  - ROC-AUC 0.73 указывает на ограниченную способность к разделению классов
- **Decision Tree**:
  - Неожиданно низкий ROC-AUC (0.67) при среднем Accuracy - признак переобучения
  - Идентичные Precision и Recall говорят о сбалансированности дерева
- **SVM (RBF)**:
  - Низкий Recall (0.60) при относительно высоком Precision (0.66) - агрессивная классификация
  - Плохой ROC-AUC (0.63) - RBF ядро плохо подходит для данных

- **Проблемные модели** (Naive Bayes, Logistic Regression, Linear SVM, SGD):
  - Крайне низкий Precision у линейных моделей (0.27) - много ложных срабатываний
  - Naive Bayes имеет высокий Precision (0.75) но очень низкий Recall (0.49) - слишком консервативен
  - ROC-AUC < 0.5 у Logistic Regression - модель хуже случайного угадывания
  - Отсутствие ROC-AUC для некоторых моделей связано с их детерминированным характером

Выводы:

- Лучшей моделью является **Random Forest** с максимальными значениями по всем метрикам
- Ансамблевые методы (Random Forest, Gradient Boosting) превосходят одиночные модели
- Линейные модели показали неудовлетворительные результаты, что говорит о нелинейной природе данных
- Низкие значения ROC-AUC у многих моделей указывают на сложность задачи классификации

3.4 Классификация CC50 $>$ медианы

Таблица 5: Метрики для CC50 $>$ медианы

Модель	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC-AUC**
Random Forest	0.81	0.81	0.81	0.81	0.89
Gradient Boosting	0.78	0.78	0.78	0.78	0.88
AdaBoost	0.79	0.79	0.79	0.79	0.85
QDA	0.77	0.77	0.77	0.77	0.85
LDA	0.76	0.76	0.76	0.76	0.83
Decision Tree	0.72	0.72	0.72	0.72	0.77
k-NN	0.66	0.66	0.66	0.66	0.69
SVM (RBF)	0.64	0.74	0.64	0.58	0.68
Logistic Regression	0.48	0.23	0.48	0.31	0.50
Naive Bayes	0.52	0.27	0.52	0.36	0.47
Linear SVM	0.48	0.23	0.48	0.31	-
SGD Classifier	0.48	0.23	0.48	0.31	-
Ridge Classifier	0.76	0.76	0.76	0.76	-

Интерпретация результатов (приложение 6):

- **Random Forest** подтверждает статус лучшей модели:
  - Исключительно высокий ROC-AUC (0.89) - выдающаяся разделяющая способность
  - Идеально сбалансированные Precision и Recall (по 0.81) - модель не имеет смещения

  - F1-score 0.81 подтверждает общую сбалансированность модели

- **Gradient Boosting** показывает:
  - Чуть более низкие показатели (Accuracy 0.78), но почти равный ROC-AUC (0.88)
  - Небольшое преимущество перед AdaBoost, особенно заметное по ROC-AUC

- **AdaBoost** демонстрирует:
  - Устойчивые результаты (Accuracy 0.79) между Gradient Boosting и QDA
  - Хороший баланс между Precision и Recall
  - ROC-AUC 0.85 - достойный результат для бустингового метода

- **QDA и LDA**:
  - QDA (0.77) незначительно превосходит LDA (0.76) по Accuracy
  - QDA показывает лучший ROC-AUC (0.85 против 0.83 у LDA) - нелинейные границы более адекватны
  - Обе модели демонстрируют стабильность (равные Precision и Recall)

- **Decision Tree**:
  - Accuracy 0.72 - существенно хуже ансамблевых методов
  - ROC-AUC 0.77 подтверждает ограничения одиночного дерева
  - Несмотря на простоту, показывает достойные результаты

- **SVM (RBF)**:
  - Явный дисбаланс: высокий Precision (0.74) при низком Recall (0.64)
  - F1-score 0.58 указывает на проблему с гармонией между ошибками
  - ROC-AUC 0.68 - RBF ядро не оптимально для данных

- **Проблемные модели**:
  - Линейные модели (Logistic Regression, Linear SVM, SGD) показывают катастрофически низкий Precision (0.23)
  - Naive Bayes имеет приемлемый Accuracy (0.52) но крайне низкий ROC-AUC (0.47) - хуже случайного угадывания
  - Ridge Classifier (0.76) показывает неожиданно хорошие результаты для линейного метода

Выводы:

- **Random Forest** остается безусловным лидером с ROC-AUC 0.89
- Ансамблевые методы занимают весь пьедестал (Random Forest, Gradient Boosting, AdaBoost)
- Квадратичный дискриминантный анализ (QDA) превзошел линейные аналоги
- Линейные методы совершенно не подходят для данной задачи
- Результаты подтверждают сложную нелинейную природу данных
- Разрыв между лучшей и худшей моделью (Accuracy=0.33) подчеркивает важность выбора алгоритма

3.5 Классификация SI > медианы

Таблица 6: Метрики для SI > медианы

Модель	Accuracy	Precision	Recall	F1	ROC-AUC
Random Forest	0.68	0.68	0.68	0.68	0.72
QDA	0.64	0.68	0.64	0.63	0.72
Gradient Boosting	0.66	0.66	0.66	0.66	0.71
AdaBoost	0.66	0.66	0.66	0.66	0.66
LDA	0.62	0.62	0.62	0.62	0.65
SVM (RBF)	0.61	0.67	0.61	0.55	0.65
k-NN	0.59	0.60	0.59	0.59	0.63
Decision Tree	0.57	0.57	0.57	0.57	0.58
Logistic Regression	0.54	0.29	0.54	0.38	0.54
Naive Bayes	0.47	0.75	0.47	0.30	0.45
Linear SVM	0.54	0.29	0.54	0.38	-
SGD Classifier	0.54	0.29	0.54	0.38	-
Ridge Classifier	0.61	0.61	0.61	0.61	-

Интерпретация результатов (приложение 7):

- **Random Forest** сохраняет лидерство, но с более скромными результатами:
  - ROC-AUC 0.72 указывает на умеренную разделяющую способность
  - Сбалансированные Precision и Recall (0.68) - модель остается нейтральной
  - F1-score 0.68 подтверждает отсутствие перекоса в ошибках
- **QDA** показывает неожиданно хорошие результаты:
  - Второе место по ROC-AUC (0.72), несмотря на Accuracy 0.64
  - Высокий Precision (0.68) при более низком Recall (0.64) - консервативная классификация
- **Gradient Boosting**:
  - Третье место по ROC-AUC (0.71)
  - Сбалансированные показатели Precision и Recall (0.66)
  - Незначительно уступает Random Forest по всем метрикам
- **AdaBoost**:
  - Такой же Accuracy (0.66), как у Gradient Boosting, но хуже ROC-AUC (0.66)
  - Менее устойчив к сложностям данных по сравнению с другими ансамблевыми методами
- **LDA и Ridge Classifier**:
  - Показывают схожие результаты (Accuracy 0.61-0.62)
  - LDA имеет чуть лучший ROC-AUC (0.65)
  - Оба метода демонстрируют стабильность (равные Precision и Recall)
- **SVM (RBF)**:

  - Явный дисбаланс: Precision 0.67 vs Recall 0.61
  - Низкий F1-score (0.55) указывает на проблемы с балансом ошибок
  - ROC-AUC 0.65 - неэффективное использование RBF ядра
- **Проблемные модели**:
  - Линейные модели (Logistic Regression, Linear SVM, SGD) демонстрируют:
    * Катастрофически низкий Precision (0.29) при среднем Recall (0.54)
    * F1-score около 0.38 - неприемлемое качество
  - Naive Bayes:
    * Аномально высокий Precision (0.75) при очень низком Recall (0.47)
    * ROC-AUC 0.45 - хуже случайного угадывания
  - Decision Tree:
    * Самый низкий ROC-AUC (0.58) среди нелинейных методов
    * Accuracy 0.57 - недостаточное качество для практического применения

Выводы:

- **Random Forest** остается лучшим выбором, но с заметно более низкими абсолютными показателями
- **QDA** неожиданно занял второе место, обойдя Gradient Boosting по ROC-AUC
- Ансамблевые методы демонстрируют относительное превосходство, но с меньшим отрывом
- Линейные методы полностью непригодны для решения задачи

3.6 Классификация SI $>$ 8

Таблица 7: Метрики для SI $>$ 8

Модель	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC-AUC**
QDA	0.68	0.70	0.68	0.68	0.71
Gradient Boosting	0.73	0.72	0.73	0.71	0.71
Random Forest	0.70	0.69	0.70	0.68	0.71
Decision Tree	0.70	0.69	0.70	0.68	0.69
LDA	0.69	0.67	0.69	0.67	0.68
AdaBoost	0.70	0.68	0.70	0.68	0.65
k-NN	0.61	0.59	0.61	0.60	0.63
SVM (RBF)	0.69	0.70	0.69	0.63	0.62
Logistic Regression	0.65	0.42	0.65	0.51	0.54
Naive Bayes	0.36	0.77	0.36	0.19	0.46
Linear SVM	0.65	0.42	0.65	0.51	-
SGD Classifier	0.65	0.42	0.65	0.51	-
Ridge Classifier	0.69	0.68	0.69	0.67	-

Интерпретация результатов (приложение 8):

- **Gradient Boosting** выходит на первое место:

  - Лучший Accuracy (0.73) среди всех моделей
  - Сбалансированные Precision (0.72) и Recall (0.73)
  - Высокий ROC-AUC (0.71), сравнимый с QDA и Random Forest
- **QDA** показывает неожиданно хорошие результаты:
  - Второе место по ROC-AUC (0.71) и F1-score (0.68)
  - Более высокий Precision (0.70) по сравнению с Recall (0.68)
  - Демонстрирует эффективность квадратичных границ решений
- **Random Forest** теряет лидерство:
  - Accuracy (0.70) ниже, чем у Gradient Boosting
  - ROC-AUC (0.71) остается на высоком уровне
  - F1-score (0.68) указывает на небольшой дисбаланс
- **Decision Tree** демонстрирует аномально хорошие результаты:
  - Accuracy (0.70) идентичен Random Forest
  - ROC-AUC (0.69) лишь немного ниже ансамблевых методов
  - Возможно, данные хорошо разделяются простыми правилами
- **SVM (RBF)** показывает противоречивые результаты:
  - Высокий Accuracy (0.69) и Precision (0.70)
  - Но низкий F1-score (0.63) и ROC-AUC (0.62)
  - Указывает на проблемы с калибровкой вероятностей
- **Проблемные модели**:
  - Naive Bayes:
    * Катастрофически низкий Recall (0.36) при высоком Precision (0.77)
    * Неприемлемый F1-score (0.19) и ROC-AUC (0.46)
  - Линейные модели (Logistic Regression, Linear SVM, SGD):
    * Низкий Precision (0.42) при среднем Recall (0.65)
    * F1-score около 0.51 - недостаточно для практического применения

Выводы:

- **Gradient Boosting** становится оптимальным выбором для этих данных
- **QDA** подтверждает свою эффективность для задач с квадратичными границами
- Ансамблевые методы (Gradient Boosting, Random Forest) сохраняют лидерство
- Простые модели (Decision Tree, LDA) показывают неожиданно хорошие результаты
- Линейные методы и Naive Bayes демонстрируют неприемлемое качество