СибГУТИ
СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ

Минцифры России

Кафедра вычислительных систем

Реализация моделей машинного обучения для задачи обнаружения мошеннических операций

Выпускная квалификационная работа бакалавра

Выполнил: студент группы ИС-142
Григорьев Юрий

Руководитель: ст. преп. Кафедры ВС
Крамаренко Константин Евгеньевич

Постановка задачи

- Цель: Реализовать и сравнить разные модели машинного обучения для задачи выявления мошеннических операций

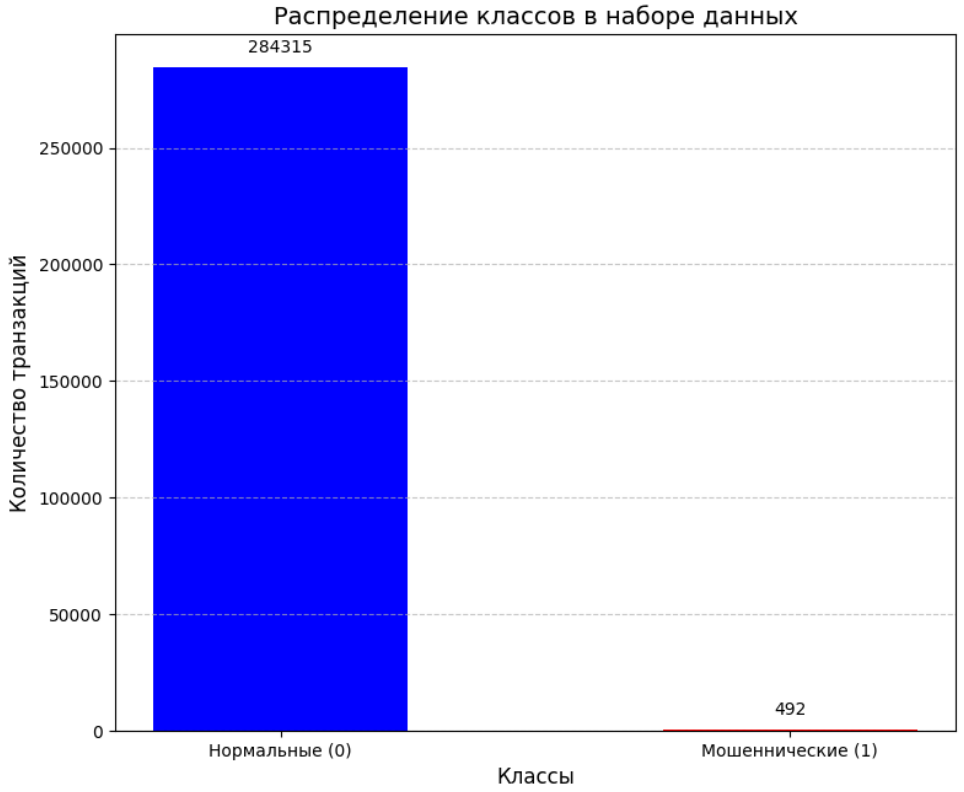
- Задачи:
  - Изучить известные решения и метрики для оценки
  - Решить проблему дисбаланса классов в наборе данных
  - Реализовать и протестировать подходящие модели
  - По наиболее важным метрикам сравнить полученные модели, определить самые точные и быстро обучаемые
  - Оценить интерпретируемость изученных моделей

Описание датасета

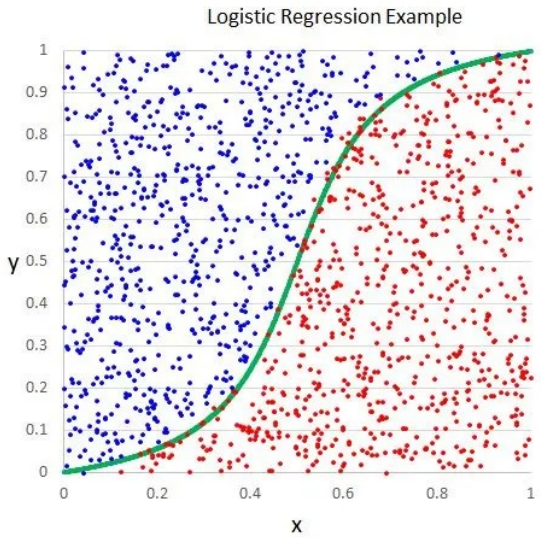
- Credit Card Fraud Detection Dataset — реальные данные европейского банка об операциях за 2 дня в сентябре 2013. 31 признак: **Time** (секунды от начала временного периода), **Amount** (сумма транзакции в денежных ед.), **Class** (0 — обычная операция, 1 — мошенническая), **V1-V28** (анонимизированные признаки), полученные с помощью метода главных компонент (*PCA*).

V1-V28 могли включать в себя номер счета, местоположение, персональную информацию, кредитный рейтинг, тип операции (например, снятие наличных, внутрибанковский перевод, покупка в магазине).

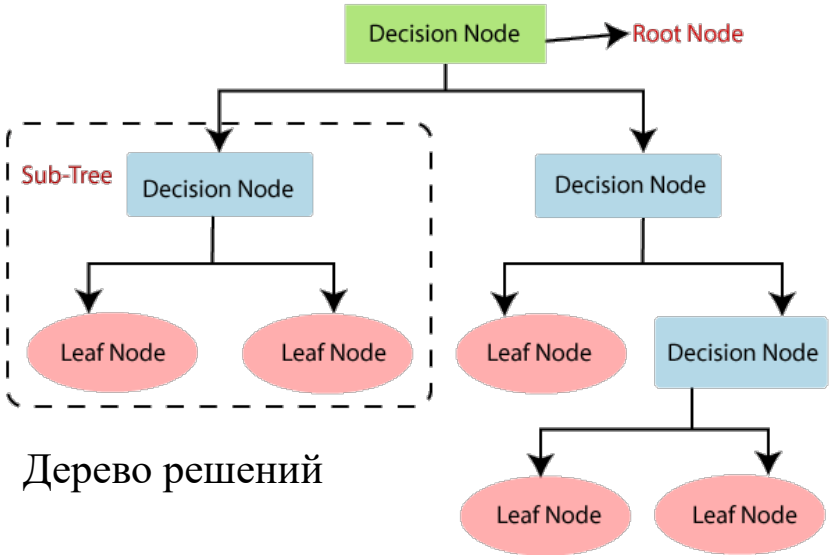
Необходима предобработка — балансировка признаков с помощью SMOTE (oversampling) и масштабирование (scaling).



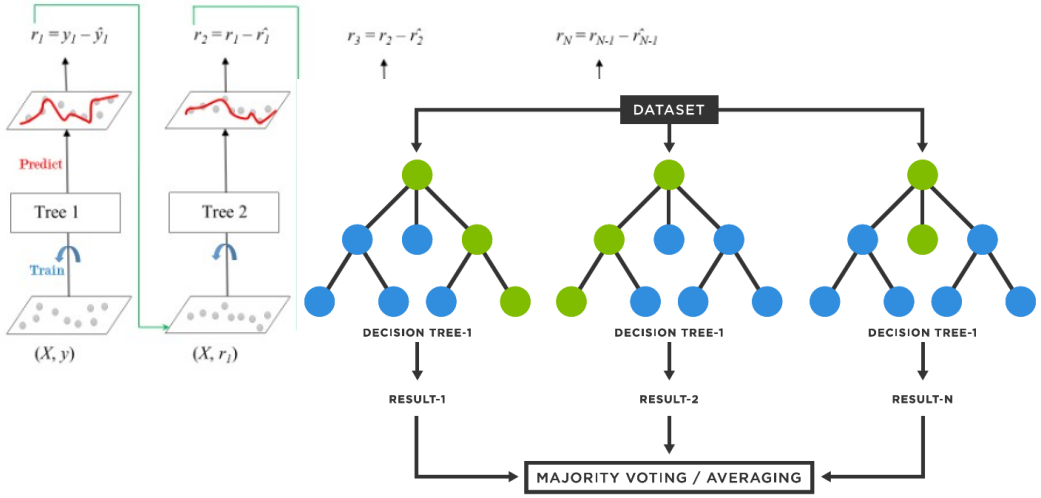
Выбранные модели



Простейшая —
логистическая регрессия

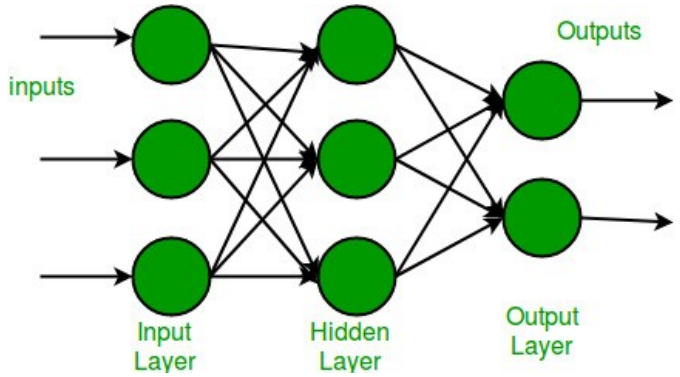


Дерево решений



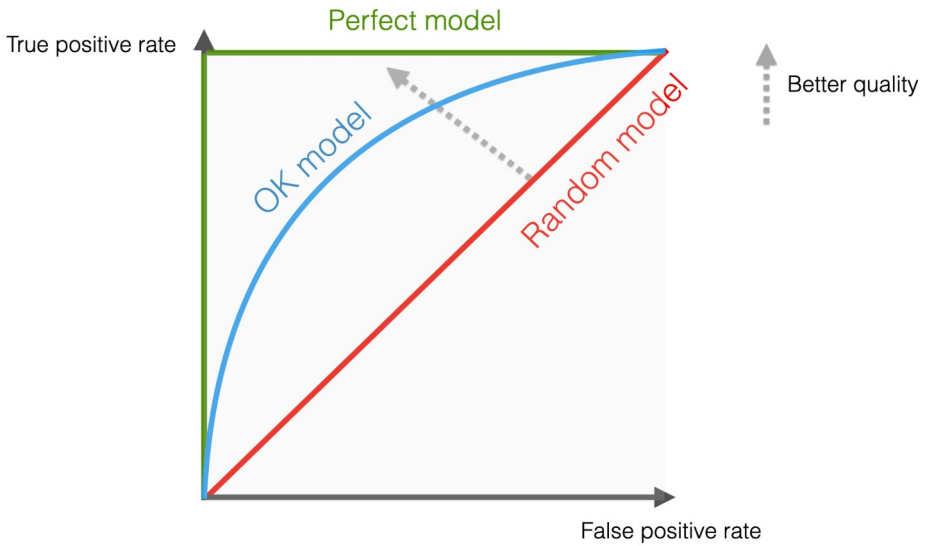
Ансамблевые, основанные на деревьях решений —
Случайный лес, Градиентный бустинг

Сложнейшая —
многослойный перцептрон, нейронная сеть



Выбранные метрики

Главная — **ROC-AUC**



0 — пропускаем слишком много положительного класса, ложно классифицируем слишком много отрицательного;
0.5 — случайное предсказание;
1 — идеально классифицируем, не даем ложноположительных результатов

Не зависит от порогового значения, как F1-score.

Дополнительная — **Recall (полнота)**

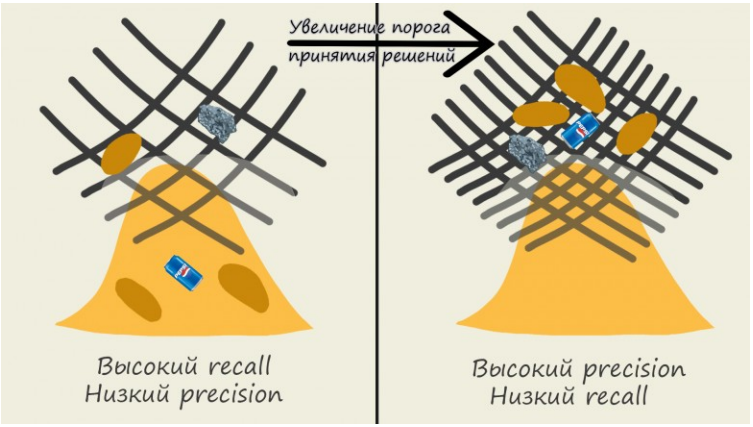
Отражает, какую долю положительного класса мы нашли. Не учитывает ложные срабатывания на нормальных транзакциях. Лучше пометить «лишние» транзакции мошенническими и проверить оператору вручную, чем пропустить истинные мошеннические.

Почему другие метрики не так полезны для этой задачи?

Accuracy — Высокая, даже если предсказать все операции как немошеннические (99.83%)

Precision — Показывает точность среди положительного класса (названные мошенническими и действительно являющиеся)

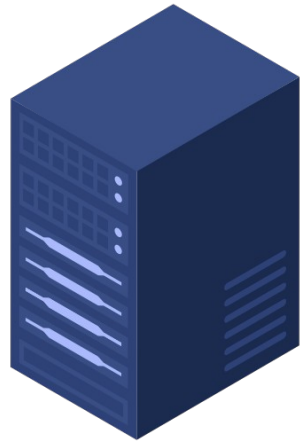
F1-score — Гармоническое среднее между Precision и Recall, зависит от выбранного порога классификации — если Precision и Recall изменятся, F1 тоже. ROC-AUC же оценивает модель независимо от порога



Тестовый стенд

Разработка и тестирование производились на ПК со следующими характеристиками:

- GPU: NVIDIA GeForce RTX 3060 (12 ГБ GDDR6 VRAM, 1882 МГц)
- CPU: 11th Gen Intel Core i5-11400F (2.6 ГГц в базовом реж.)
- RAM: 16 ГБ (2666 МГц, два канала)

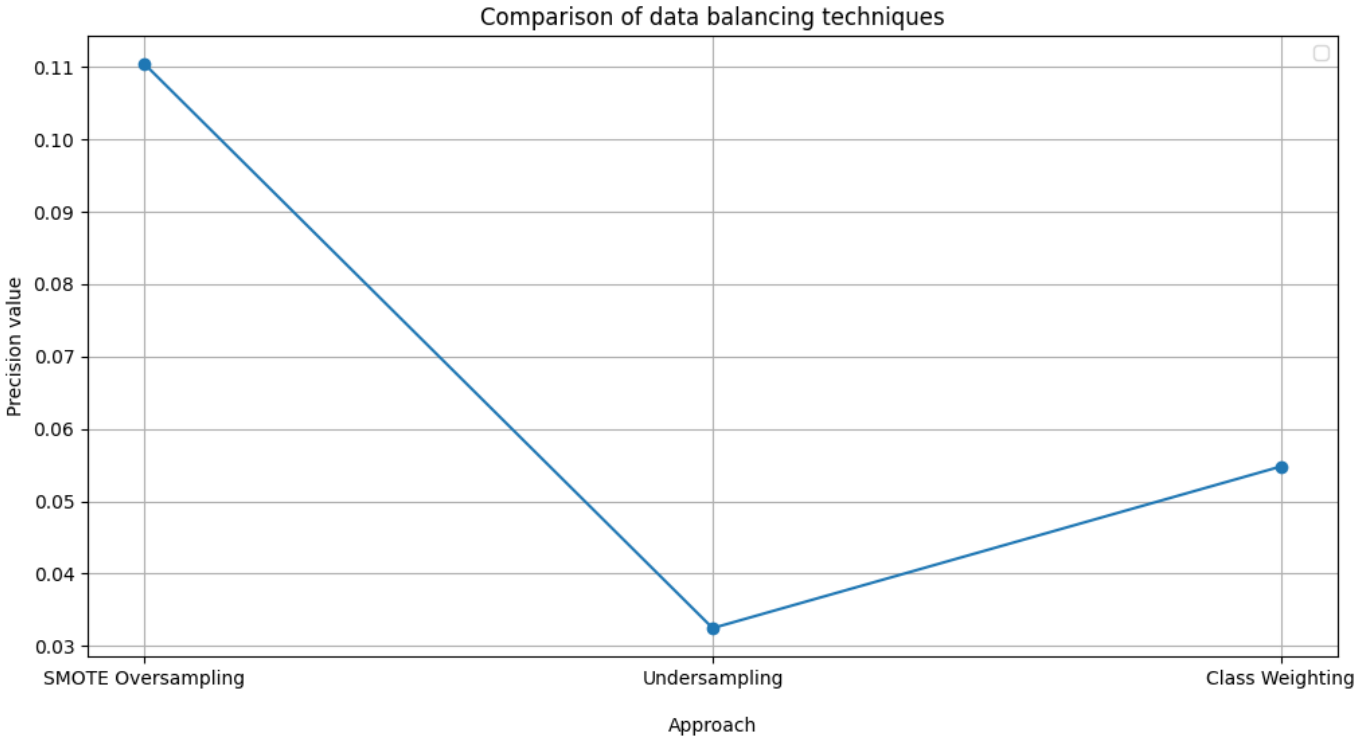


ПО:

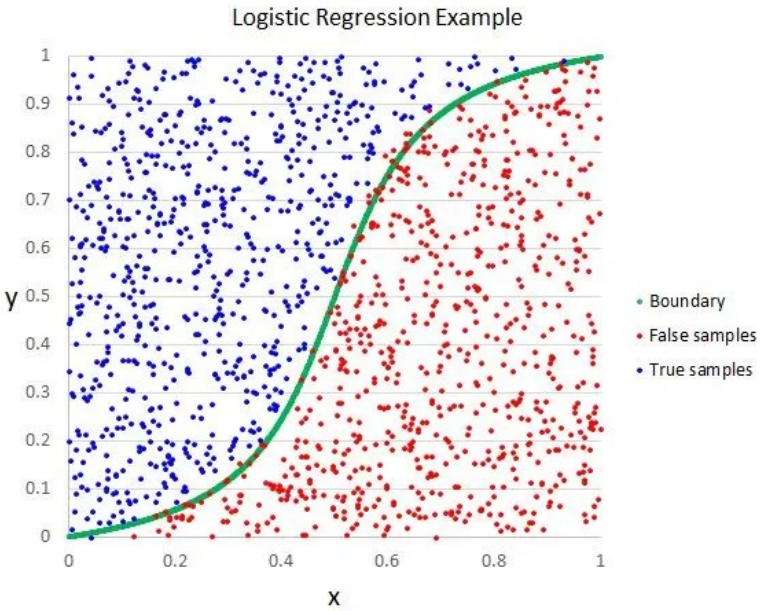
- Windows 11 версии 10.0.26100
- Python 3.13.2, включая библиотеки scikit-learn, pytorch, numpy, matplotlib, pandas и imbalanced-learn
- Драйвер NVIDIA версии 572.42 от 13.02.2025

Балансировка данных

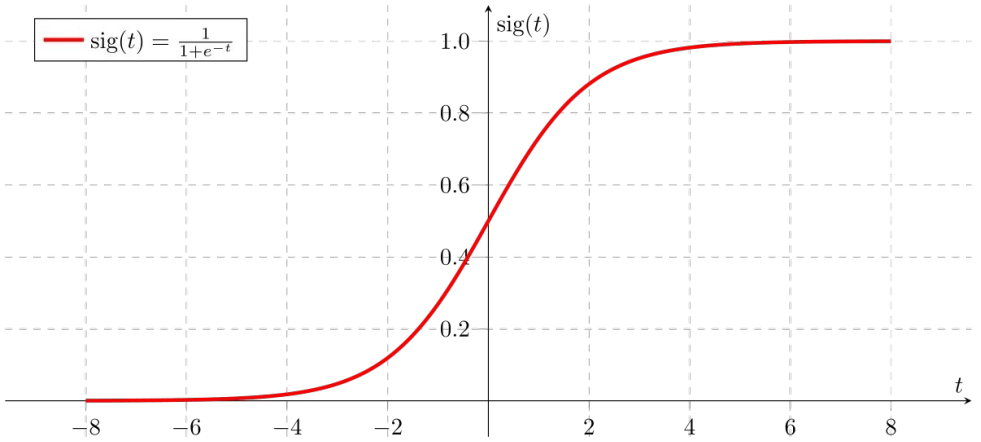
Лучшим подходом балансировки данных после тестирования оказался **SMOTE** (*oversampling*). По сравнению с **Undersampling** (*уменьшение мажоритарного класса*) и **Перевзвешиванием** (*присвоение миноритарному классу большего веса в функции потерь*) в рамках ограниченного времени результаты метрики Precision (*точность*) с его использованием превосходят конкурентные подходы более чем в 2 раза.



Логистическая регрессия



Пример работы модели для задачи бинарной классификации

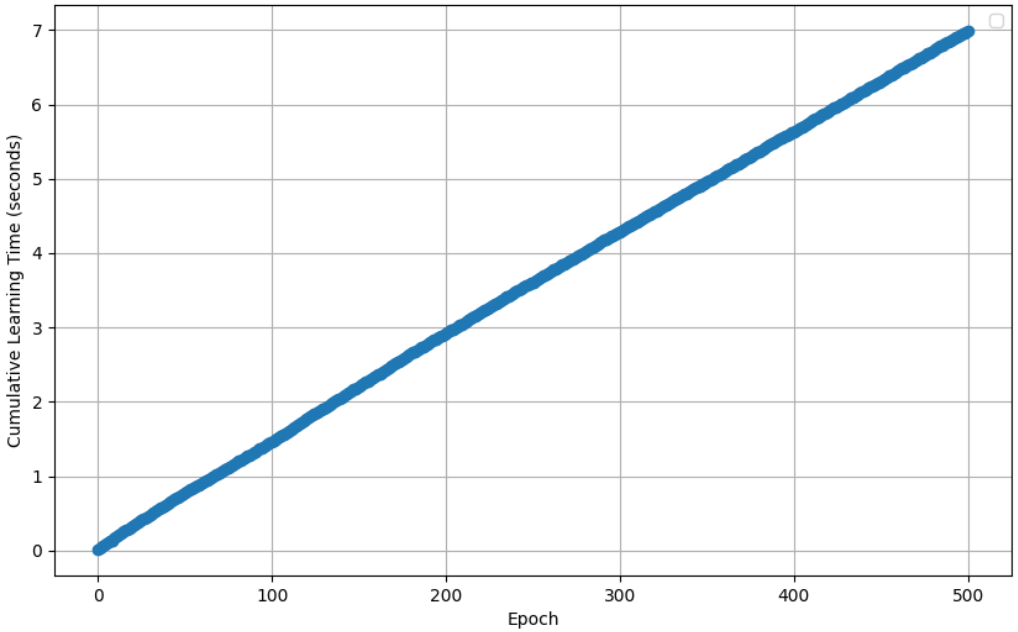


Логистическая функция (сигмоидная кривая)

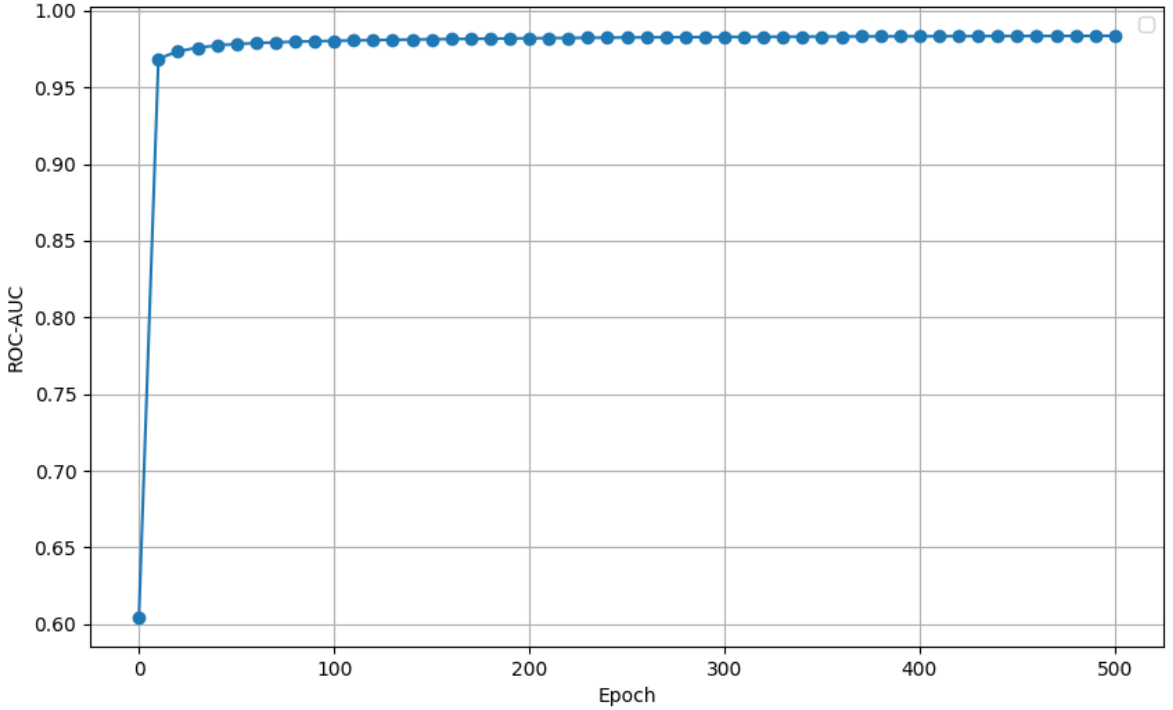
Классический метод бинарной классификации, оценивает вероятность принадлежности класса с использованием логистической функции.

Цель — оптимизировать параметры модели (веса) для минимизации функции потерь. Просто реализуема, легко интерпретируема. Предполагает линейную разделимость классов, что ограничивает эффективность при наличии сложных нелинейных зависимостей в данных.

Логистическая регрессия



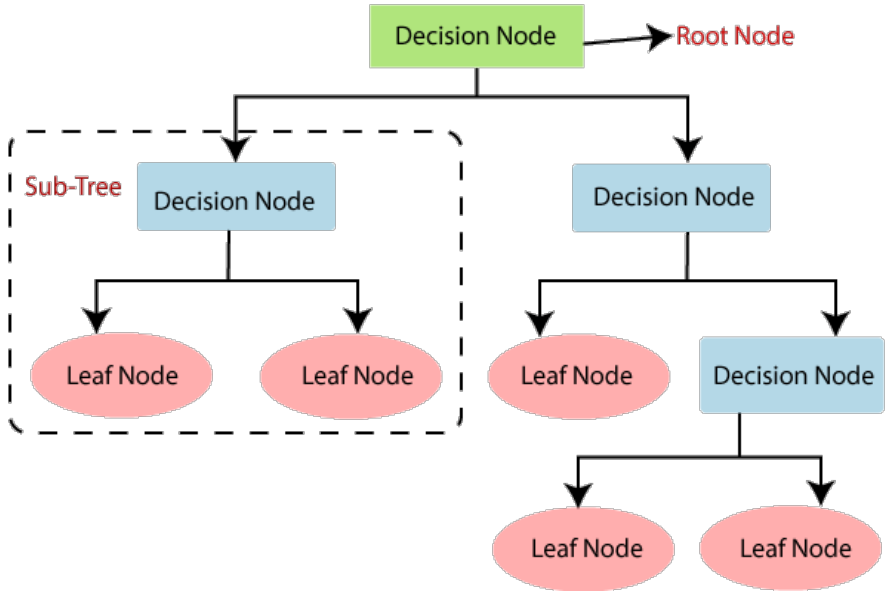
Время обучения линейно зависит от количества эпох обучения



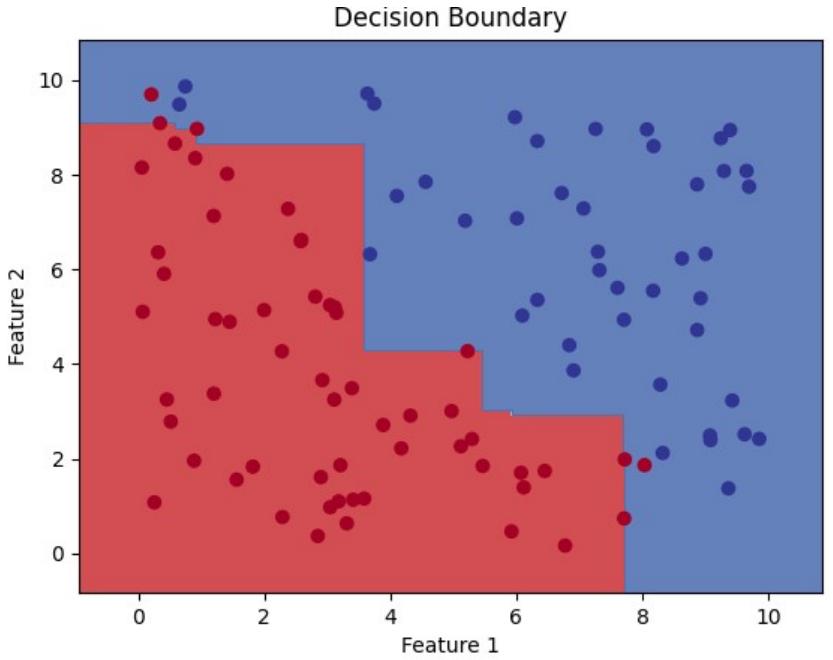
Результат ROC-AUC: 0.9848
Recall: 0.9046 (низкий)
F1-score: 0.9422

Линейная природа модели ограничивает ее способность улавливать сложные зависимости, однако обеспечивает высокую интерпретируемость и скорость обучения.

Дерево решений



Структура дерева решений

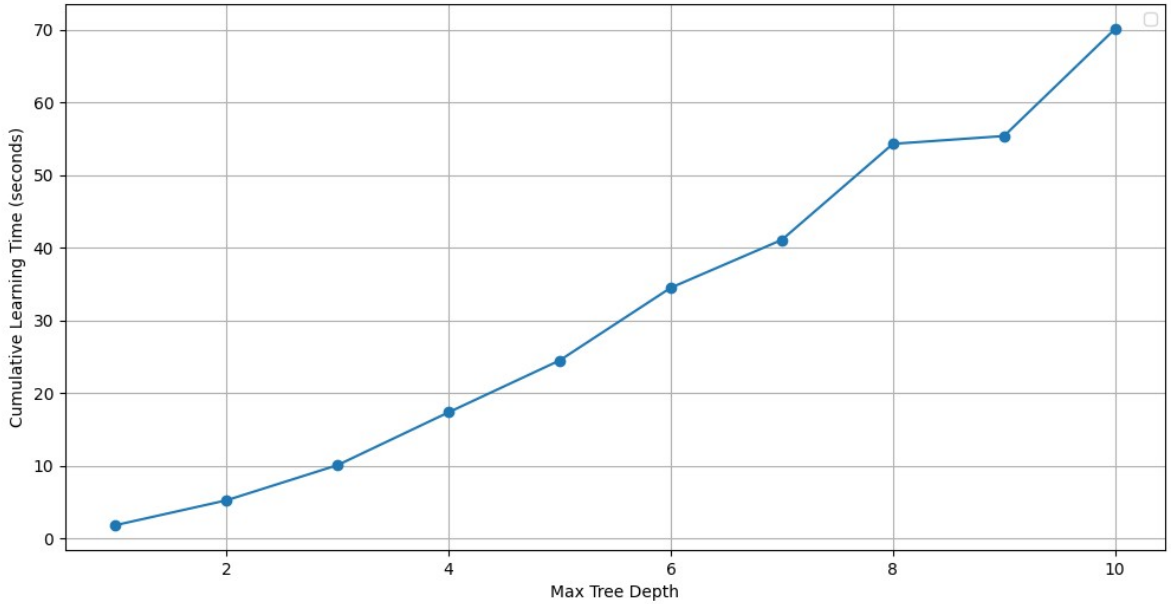


Пример работы для задачи бинарной классификации

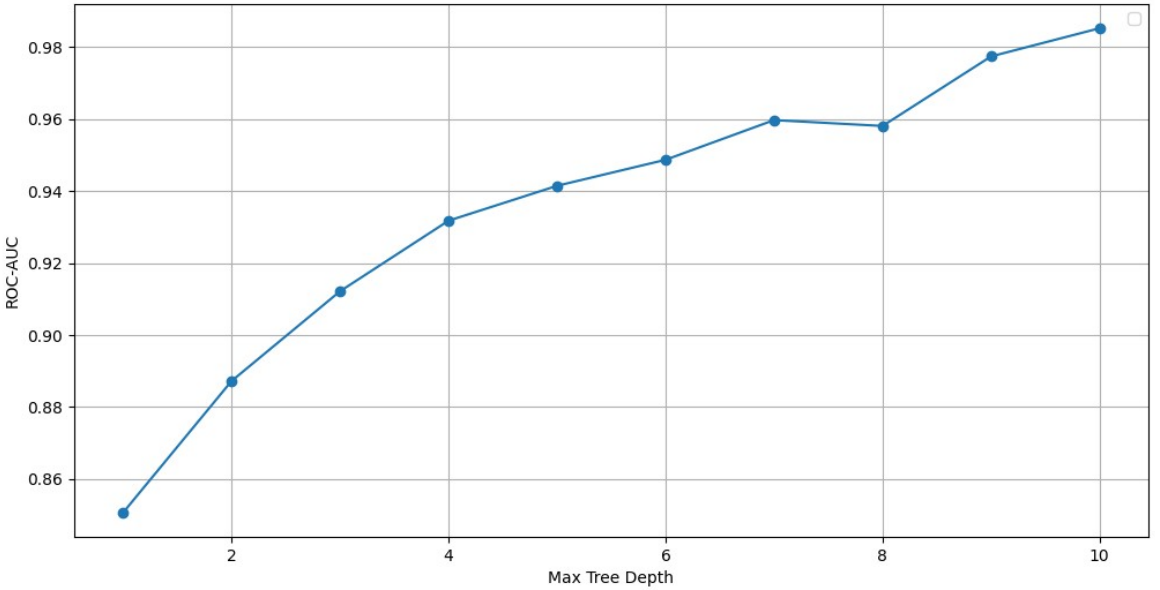
Каждый внутренний узел соответствует проверке значения определенного признака, ветви — возможным исходам этой проверки, а листья — предсказанным классам. Позволяет легко оценить вклад каждого признака.

Склонен к переобучению, особенно при отсутствии ограничений на глубину, что снижает обобщающую способность на новых данных. Нестабильно к шуму: небольшие изменения в обучающей выборке могут существенно изменить структуру.

Дерево решений



Время обучения линейно зависит от максимальной глубины дерева



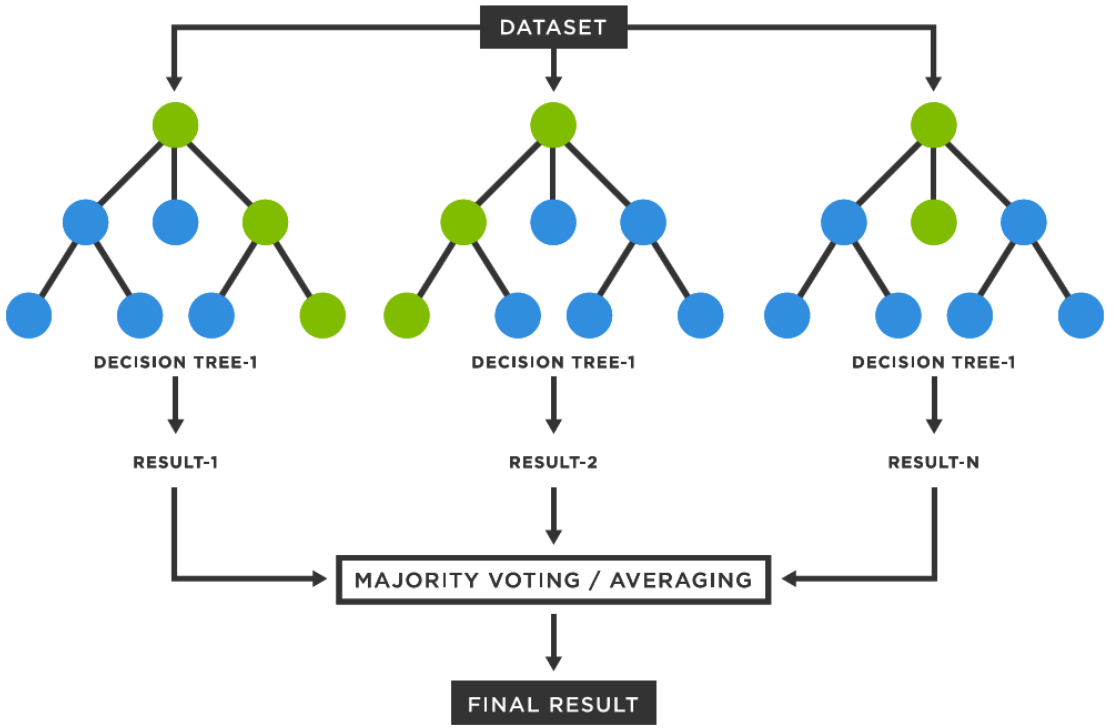
ROC-AUC логарифмически зависит от максимальной глубины дерева

Результат ROC-AUC: 0.9349
Recall: 0.9235 (лучше)
F1-score: 0.9343

Имеет большое число ложных срабатываний, качество предсказаний сильно зависит от глубины дерева. Склонность к переобучению снижает обобщающую способность.
Нужны ансамбли деревьев

Случайный лес

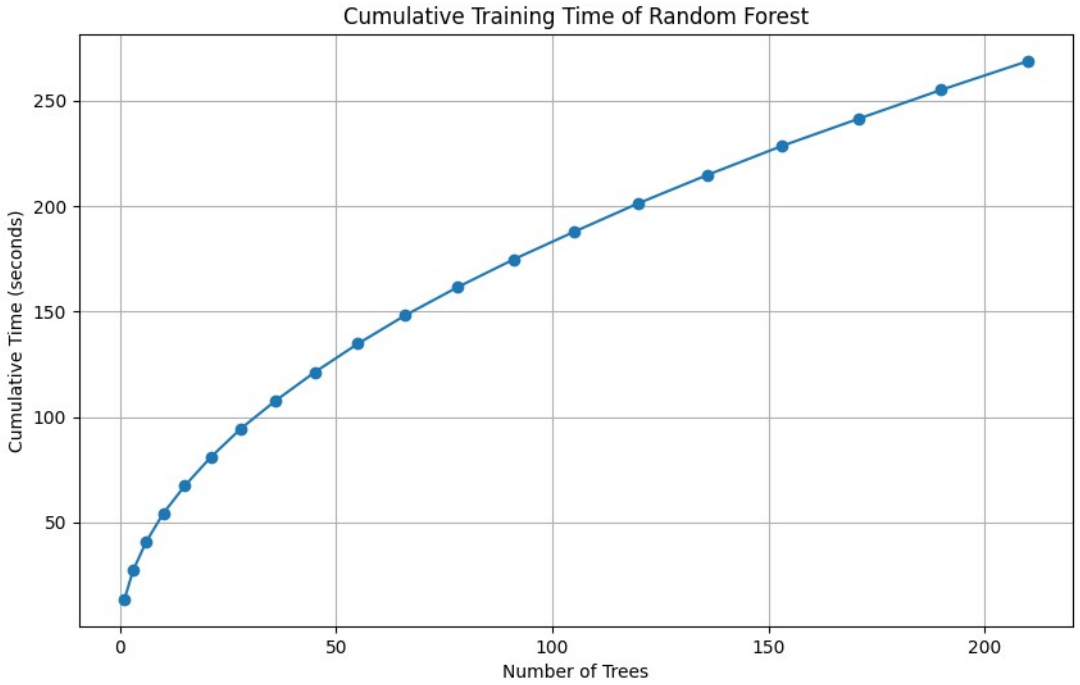
Типовая структура



Ансамблевый метод, объединяющий множество деревьев решений.
Заключается в усреднении предсказаний независимых деревьев, каждое из которых обучается на случайной подвыборке данных и признаков.

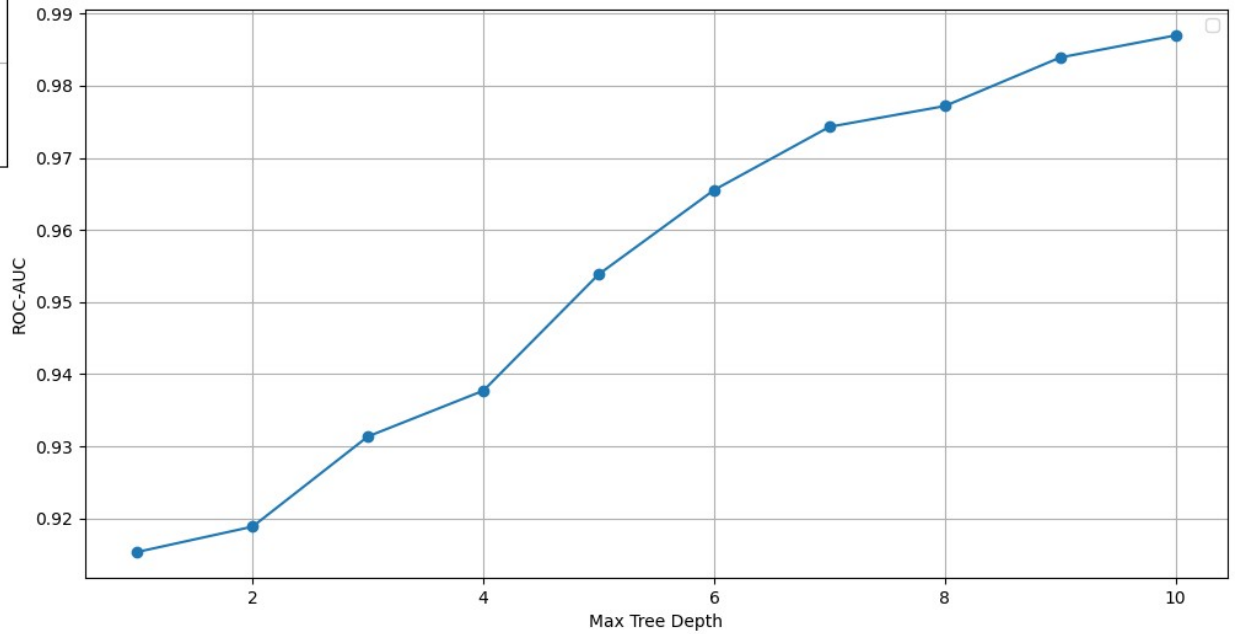
Сочетает технику bootstrap-семплирования (выборки с возвращением) и случайный выбор подмножества признаков на каждом этапе разбиения. Эффективен для задач с шумными или несбалансированными данными.

Случайный лес



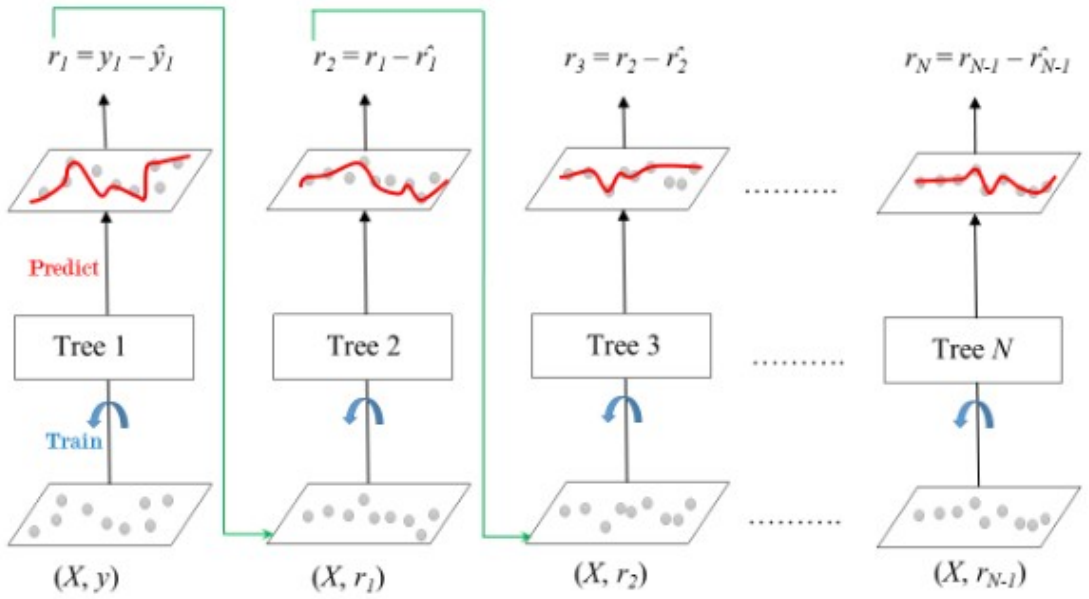
Глубина деревьев влияет на время обучения так же, как для Дерева решений

Результат ROC-AUC: 0.9536
F1-score: 0.9519



ROC-AUC от глубины деревьев решений

Градиентный бустинг

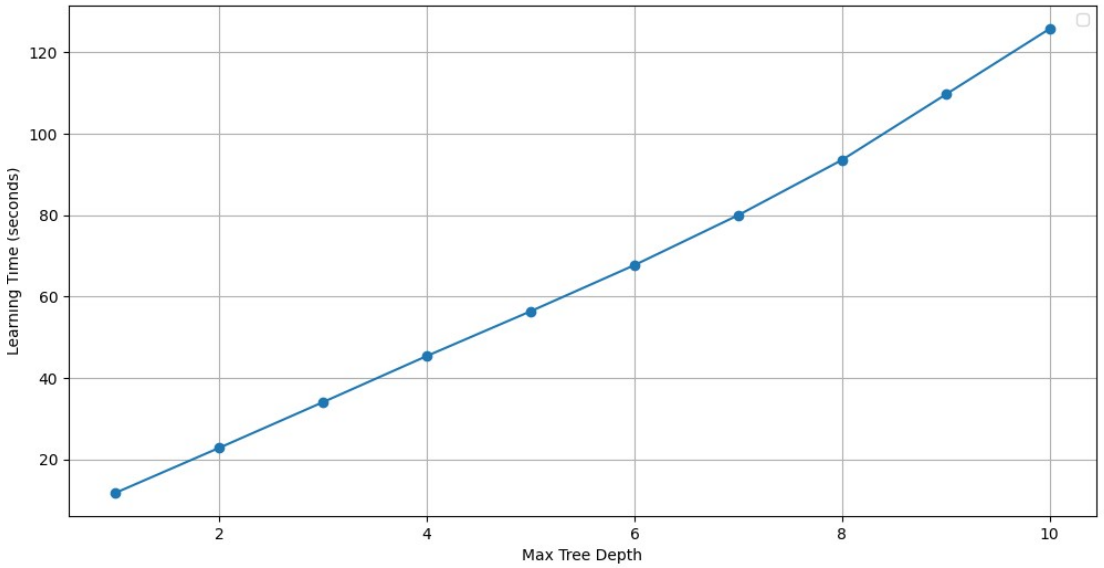


Типовая структура

Ансамблевый метод, последовательно объединяющий деревья решений для повышения точности классификации.

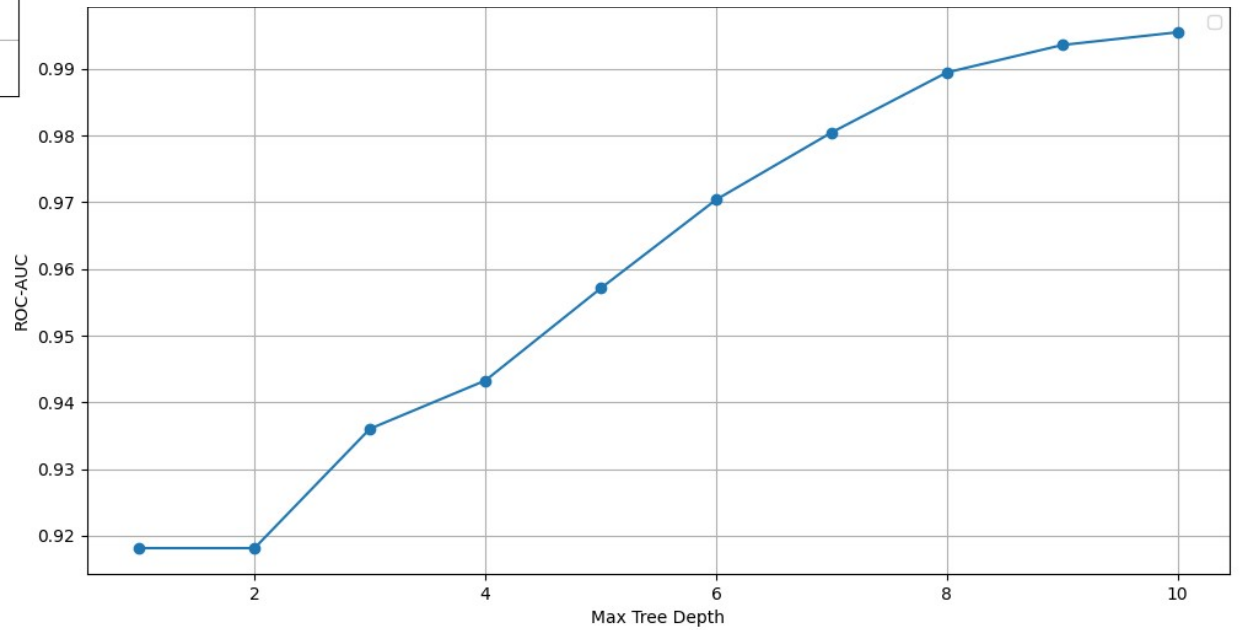
В отличие от случайного леса, где деревья обучаются независимо, в градиентном бустинге каждое следующее дерево корректирует ошибки предыдущих, минимизируя функцию потерь.

Градиентный бустинг



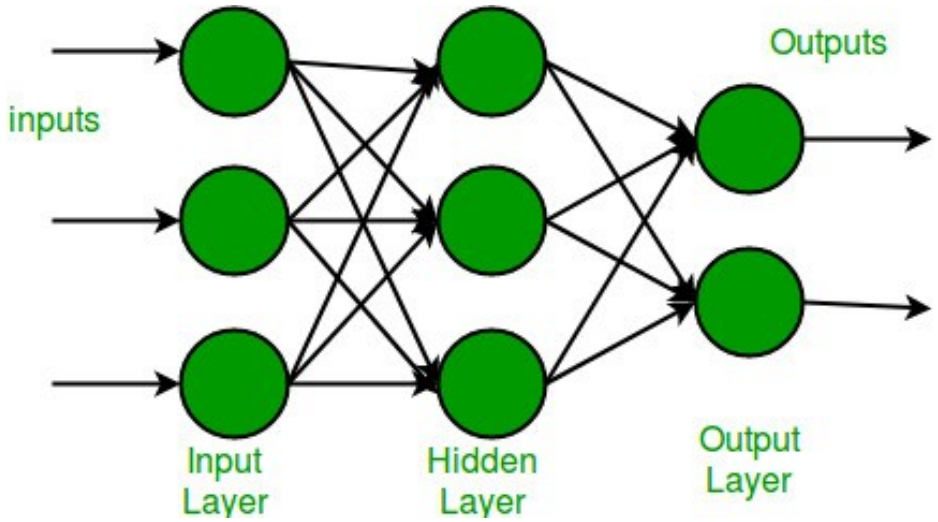
Время обучения линейно зависит от глубины деревьев решений

Гибко обучается, имеет высокую точность, но требует значительного времени для обучения

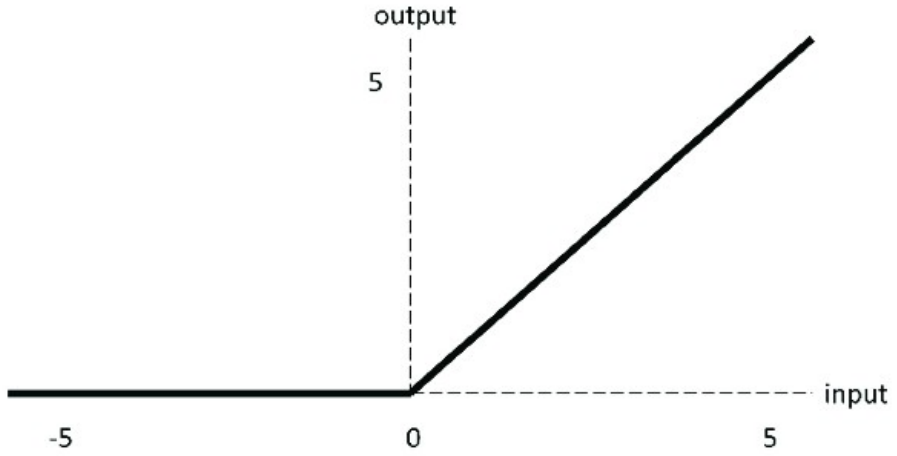


ROC-AUC от глубины деревьев решений

Многослойный перцептрон



Типовая структура



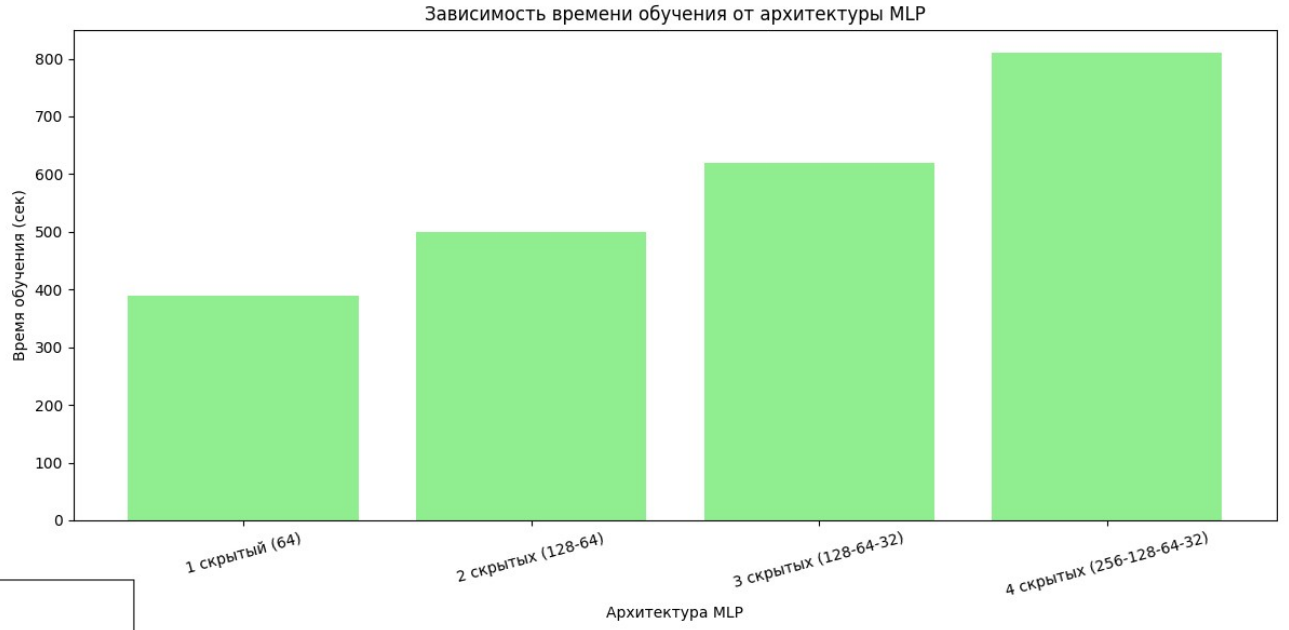
Функция активации — Rectified Linear Unit (*ReLU*)

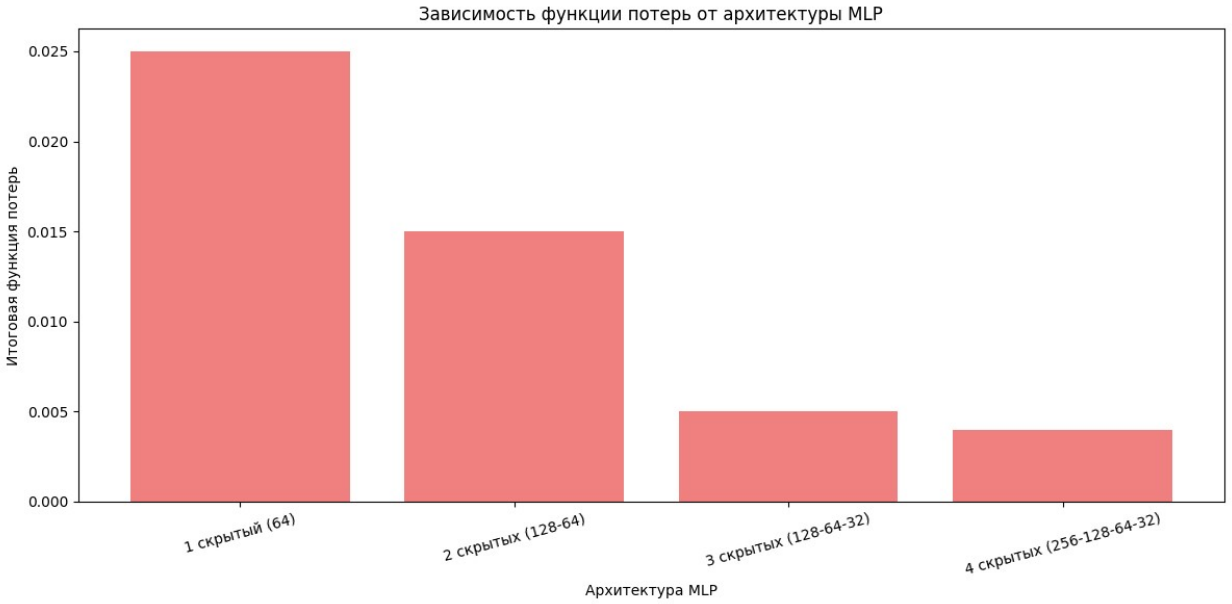
Нейронная сеть прямого распространения.
Подходит для сложных задач с большим объемом параметров и данных, но требует значительных ресурсов и плохо интерпретируется.

Каждый нейрон выполняет линейное преобразование входных данных с последующим применением нелинейной функции активации, веса корректируются от полученной функции потерь.

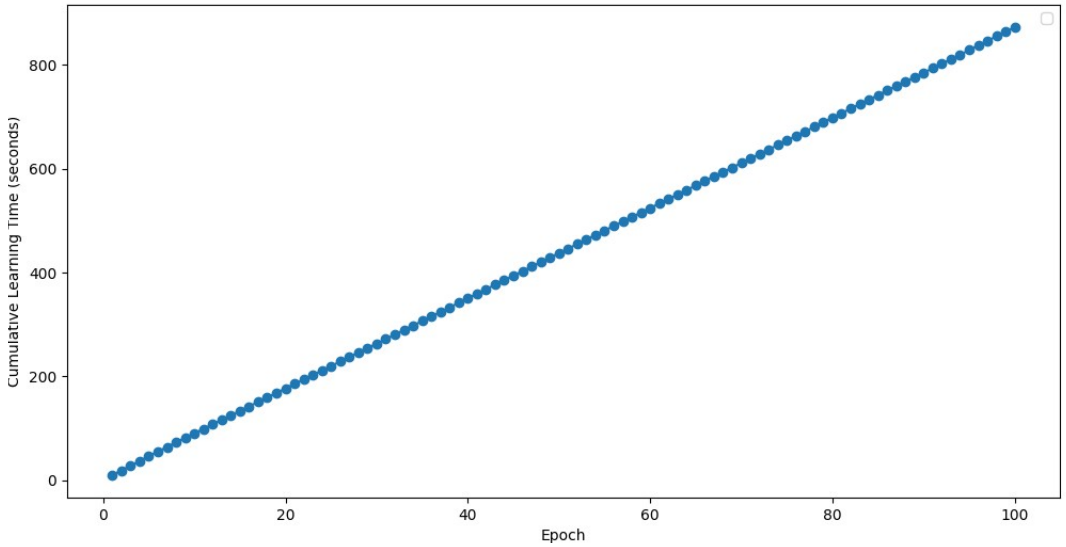
Многослойный перцептрон

Выбранная архитектура — 3 скрытых слоя (128-64-32 нейрона)





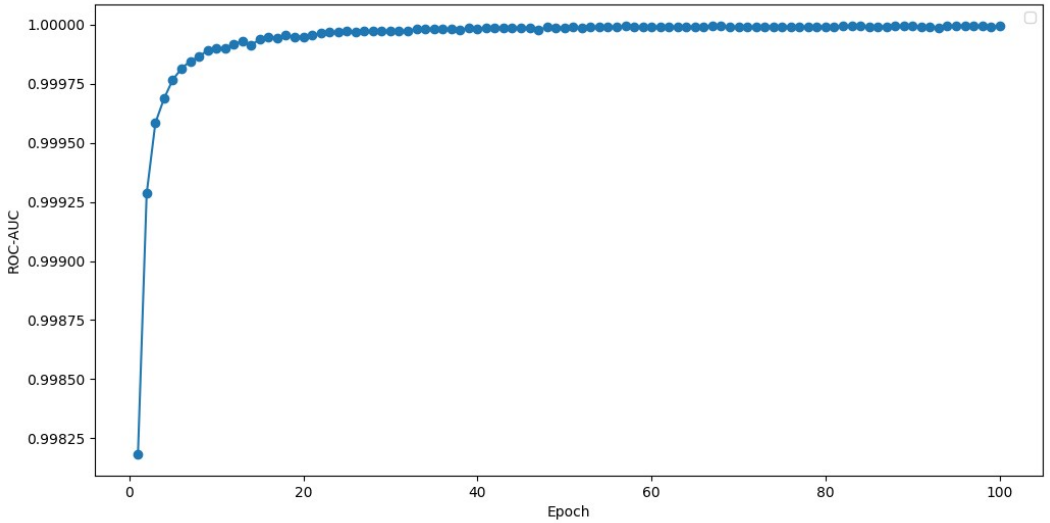
Многослойный перцептрон



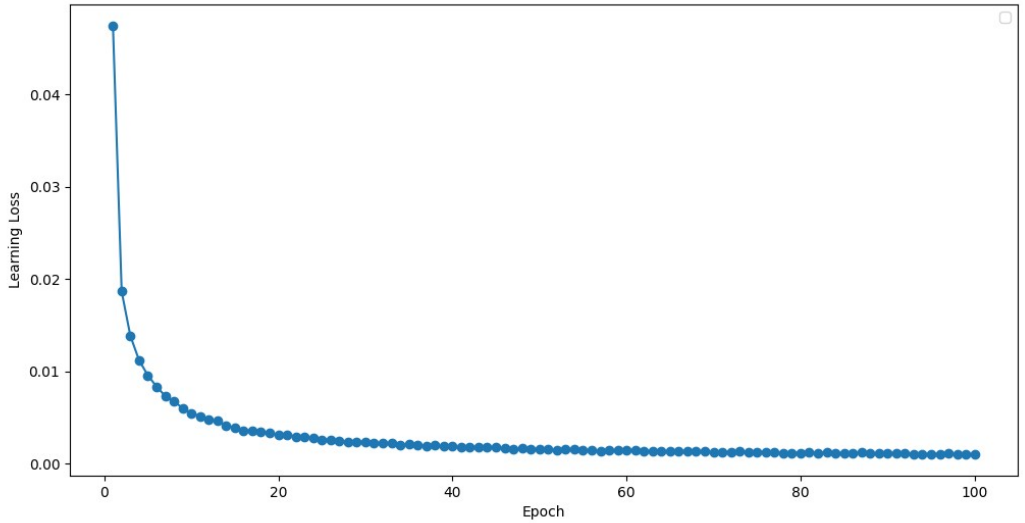
Время обучения линейно зависит от числа пройденных эпох

Графики ROC-AUC и функции потерь «отражают» друг друга

Результат ROC-AUC: 0.9998
Recall: 1.00
F1-score: 0.9996

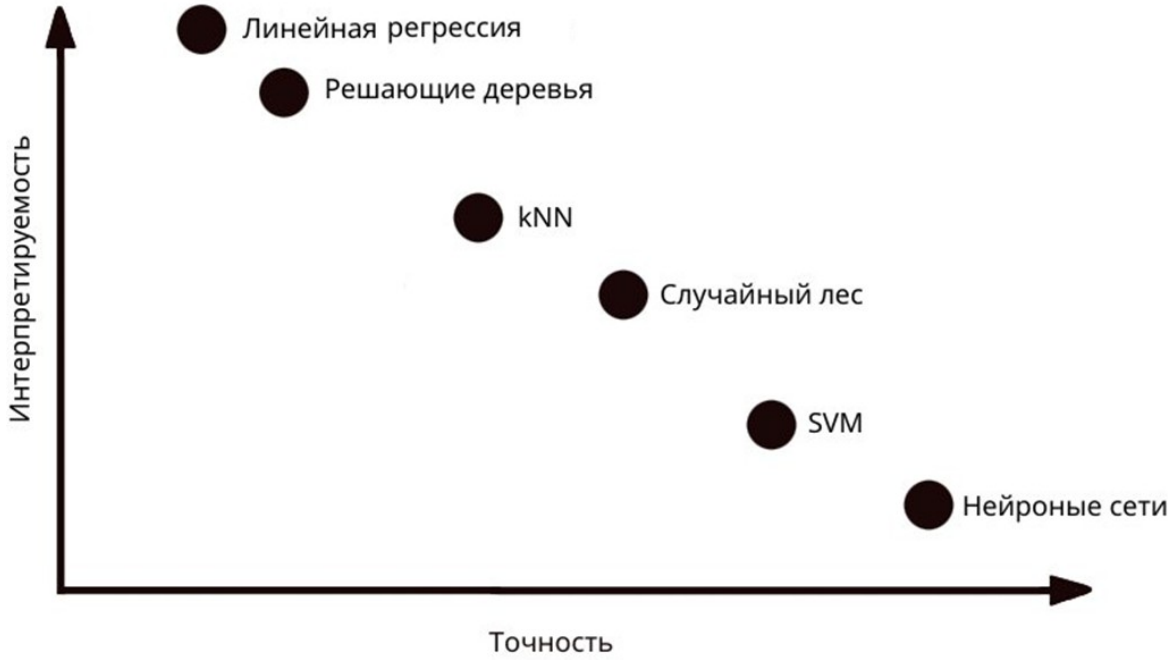


ROC-AUC от числа эпох



Функция потерь от числа эпох

Интерпретируемость

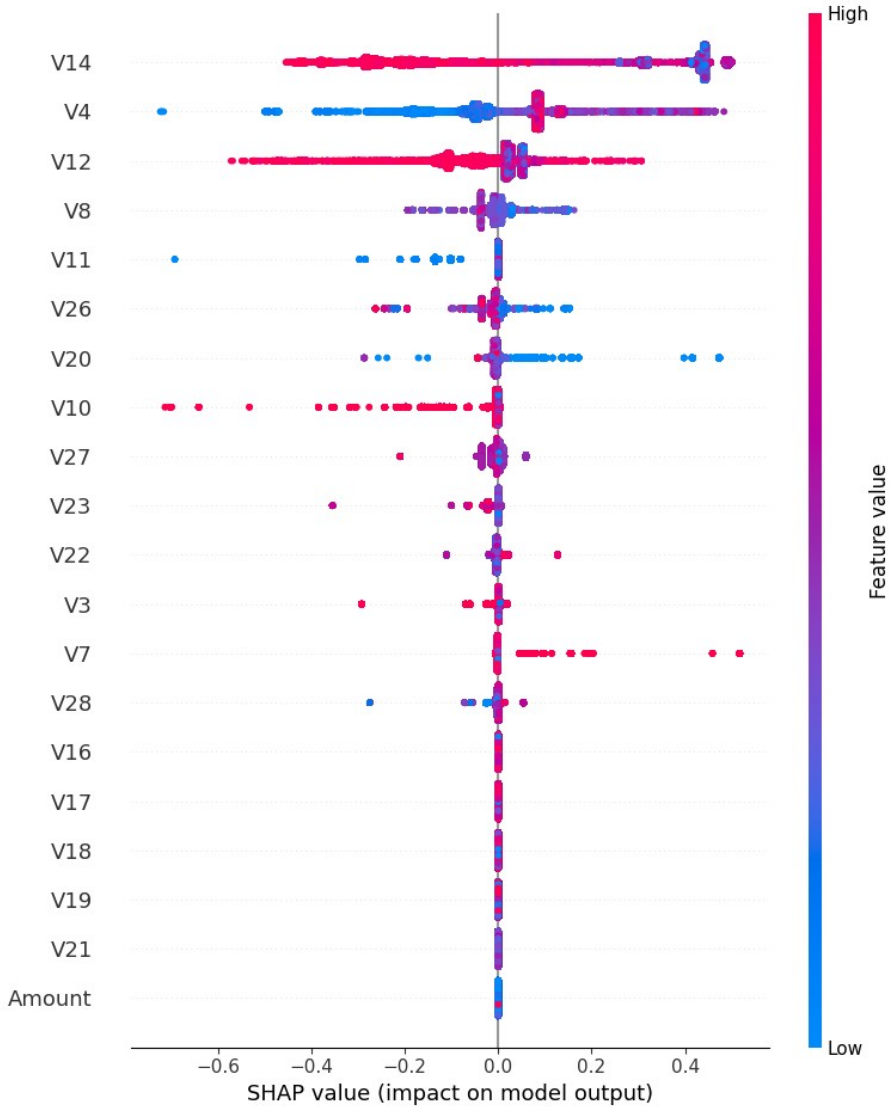


Методы интерпретации:

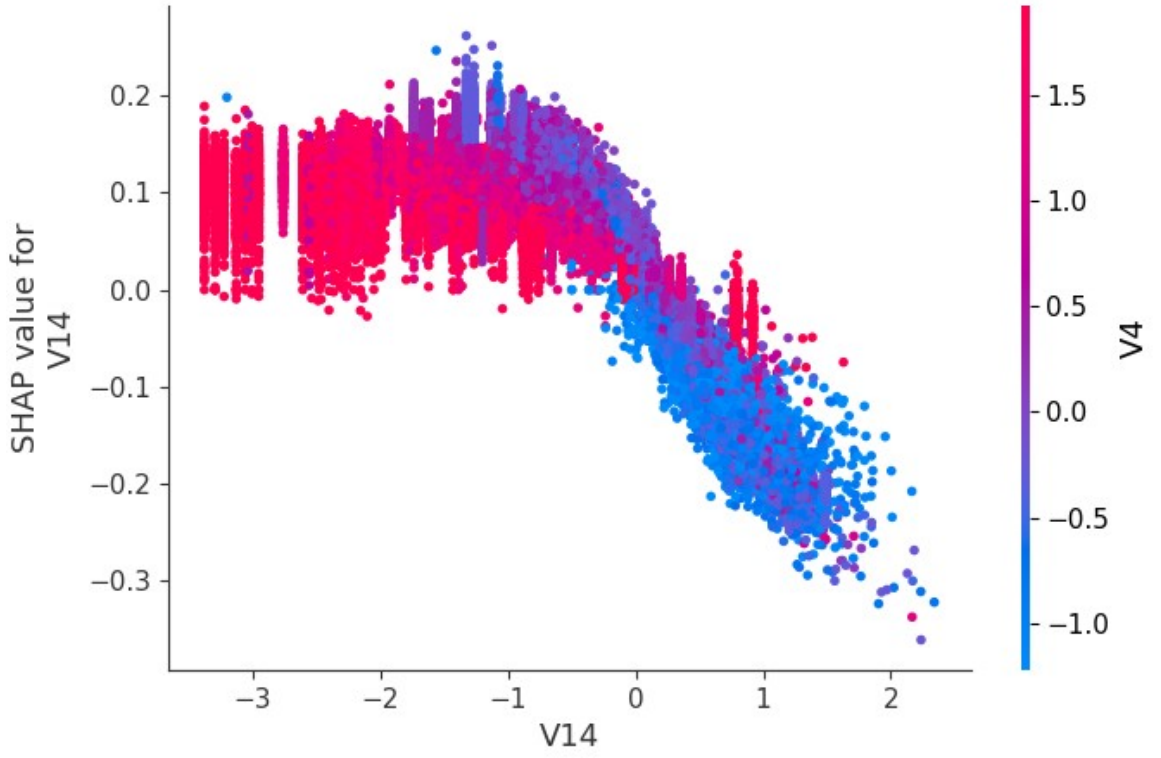
- SHAP (*Shapley Additive Explanations*) — Оценивает вклад каждого признака в предсказание (осн. на Теории Игр), обеспечивает глобальную интерпретацию.
- LIME (*Local Interpretable Model-agnostic Explanations*) — Объясняет предсказания путем аппроксимации сложной модели некоторой локально интерпретируемой функцией, выявляя ключевые зависимости в признаках для отдельно взятого (локального) примера

Интерпретируемость

SHAP

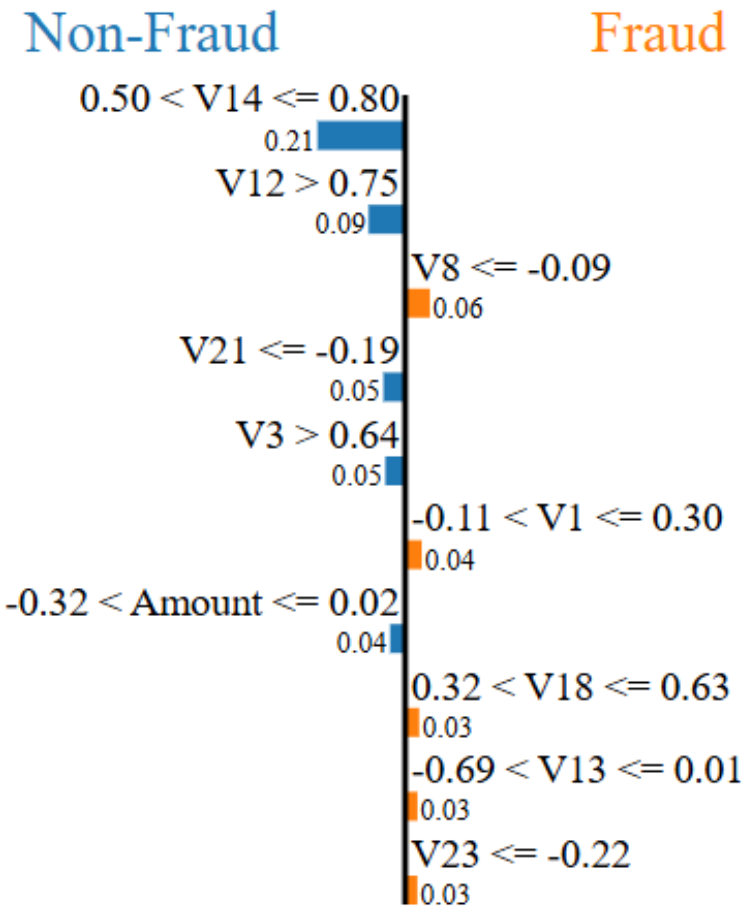


Дерево решений — зависимость SHAP от признака V14 с наложением признака V4



Интерпретируемость

LIME для модели Многослойного перцептрона



Feature	Value
V14	0.66
V12	0.84
V8	-0.16
V21	-0.23
V3	0.87
V1	0.21
Amount	-0.05
V18	0.41
V13	-0.47
V23	-0.26

- Признаки с самым большим влиянием в положительную (*оранжевый*) или отрицательную (*синий*) сторону на класс «Fraud» (мошенническая транзакция)

Заключение

Выводы:

- Проведен анализ, реализация и тестирование моделей машинного обучения для конкретного набора данных (задачи)
- Проведено сравнение разных методов балансировки данных для задачи дисбаланса классов, лучшим оказался SMOTE (oversampling), повысив производительность всех моделей
- Самый точный результат продемонстрировали модели Градиентного бустинга (ROC-AUC 0.997) и Многослойного перцептрона (ROC-AUC 0.9998). Последний является также более быстрым и во многих случаях более точным
- Проведен анализ интерпретируемости полученных результатов методами SHAP и LIME, построение обратной цепочки к признакам, которые внесли наибольший вклад в решения отдельных моделей

Заключение

Ограничения и рекомендации:

- Зависимость от конкретного датасета может влиять на «вливание» новых данных в модели и их итоговые показатели — в дальнейших исследованиях рекомендуется компоновать несколько датасетов и оценить универсальность изученных моделей
- Синтетические данные от метода SMOTE могут искажать распределение в реальных данных
- Охвачены не самые передовые методы/модели классификации данных, в дальнейших исследованиях рекомендуется рассмотреть современные нейронные сети (сверточные/трансформеры) и ансамбли (модели стекинга/блендинга) для повышения качества классификации

СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ
СибГУТИ

Минцифры России

Спасибо за внимание!