Лабораторная работа

> ## Методы регрессионного анализа

Работу подготовили:

> Панов Олег, Михаил Бабушкин, Денис Чашин, Анатолий Мезенов, Никита Боровик

Постановка задачи

В качестве задания требуется провести регрессионный анализ данных, для того чтобы оценить значение **целевой** переменной на основе **факторных**.

Полученные результаты всех методов **сравним** между собой.

> Грубо говоря, натренировать модели регрессионного анализа данных, выбрать **лучшую** из ~~худших~~ полученных моделей и показать полученные результаты.

Входные данные

> Тренировочных выборок: **5**
> Тестовых выборок: **10**

Данные выборки содержат информацию о **комментариях** в социальной сети **Facebook** ~~запрещена на территории РФ~~.

- Размер всего датасета: **0.5GB**
- Размер тренировочного датасета **602808** строк
- Уникальных строк в тренировочном датасете **602400**
- Размер тестового датасета **1089** строк

Атрибуты (*переменные*)

Датасет содержит **54** переменные.

Факторными переменными являются:

- Количество просмотров
- Длина поста
- Категория поста
- День недели
- Время дня
- *и т.д.*

Целевые значения

Одна из переменных является целевой - **Target Variable**

> Это количество комментариев под постом в следующие **H часов**

H часов - это факторная переменная!!!

Разведочный анализ

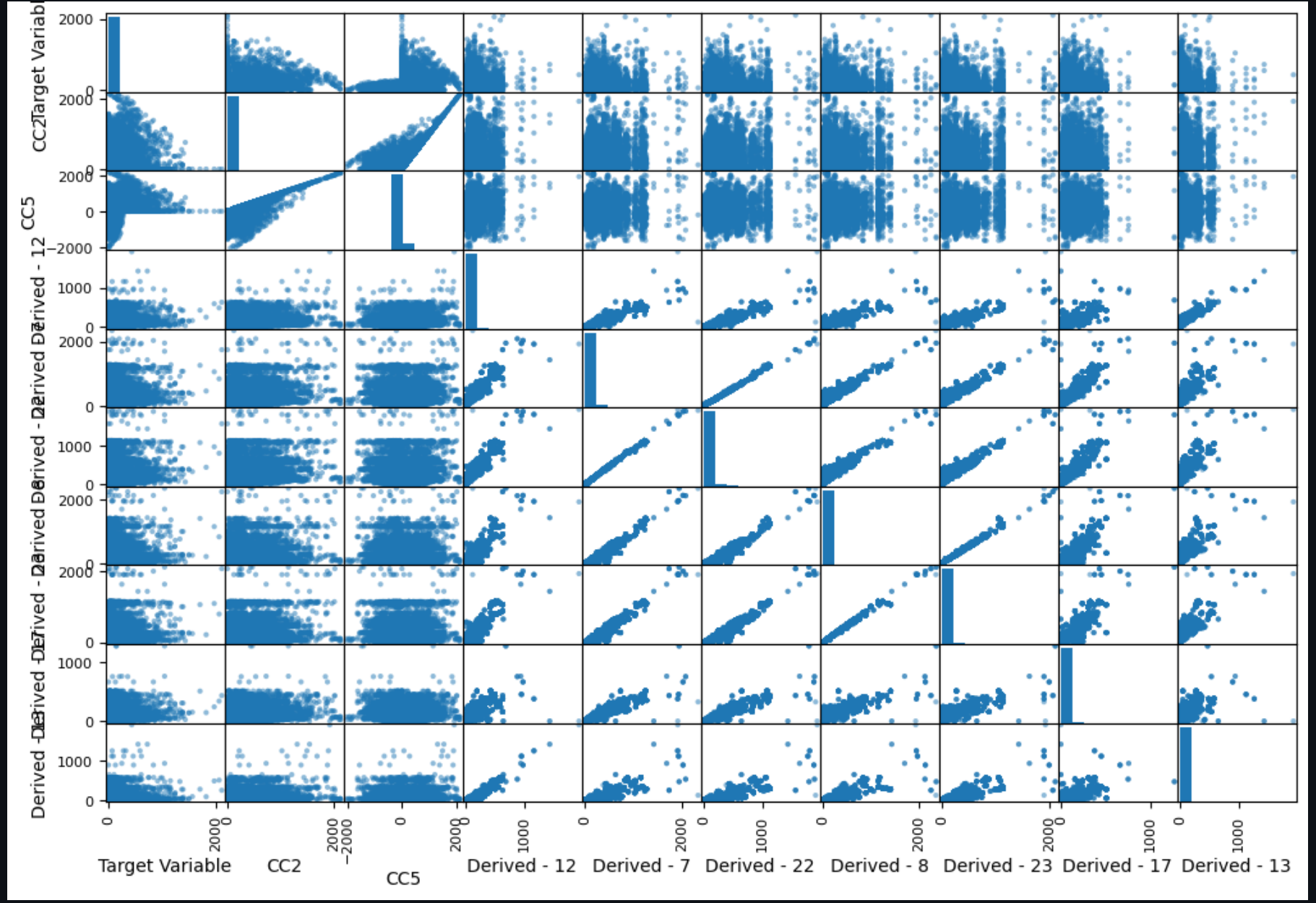
- Это процесс анализа данных в целях выявления основных характеристик, закономерностей, аномалий и тенденций, с использованием различных графических и статистических методов.
- Основная цель разведочного анализа - получить более глубокое понимание структуры данных, выделить ключевые аспекты и гипотезы, которые могут быть дальше исследованы или применены в анализе данных.

Корреляция величин

Variable	Correlation
Target Variable	1.000000
CC2	0.537412
CC5	0.372872
Derived - 12	0.366997
Derived - 7	0.357139
Derived - 22	0.354598
Derived - 8	0.350421

Матрица рассеяния

- Это **графическое представление**, в котором каждая пара переменных в наборе данных представлена в виде диаграммы рассеяния (**scatter plot**).
- Матрица рассеяния используется для **визуального исследования связей** и корреляций между парами переменных, позволяя аналитику быстро выявить потенциальные зависимости и тенденции в данных.
- Каждая ячейка матрицы содержит график, показывающий, как взаимодействуют две конкретные переменные, что помогает выявить структуры и паттерны в данных.



Матрица рассеяния

Выбор моделей

Для решения проблемы регрессии мы решили выбрать следующие модели, и распределили их между собой.

- **LS** - Михаил Бабушкин
- **Ridge** - Анатолий Мезенов
- **DT** - Никита Боровик
- **RF** - Никита Боровик
- **KNN** - Олег Панов
- **GB** - Денис Бабушкин

> И мы расскажем о них подробнее, но сначала

Напомним вам, насколько важен GridSearch

GridSearchCV – это очень мощный инструмент для автоматического подбирания параметров для моделей машинного обучения. Метод **поиска по сетке** находит наилучшую комбинацию параметров, которые дают **наименьшую ошибку**, путем обычного перебора: **он создает модель для каждой возможной комбинации параметров**.

> Ну а теперь расскажем о важном...

CV *или же кросс-валидация*

> И какую же мы используем 🔥

Кросс-валидация работает путем разделения набора данных на несколько поднаборов, называемых **фолдами**. Затем модель обучается и тестируется несколько раз, каждый раз используя **разные фолды** для тестирования и обучения.

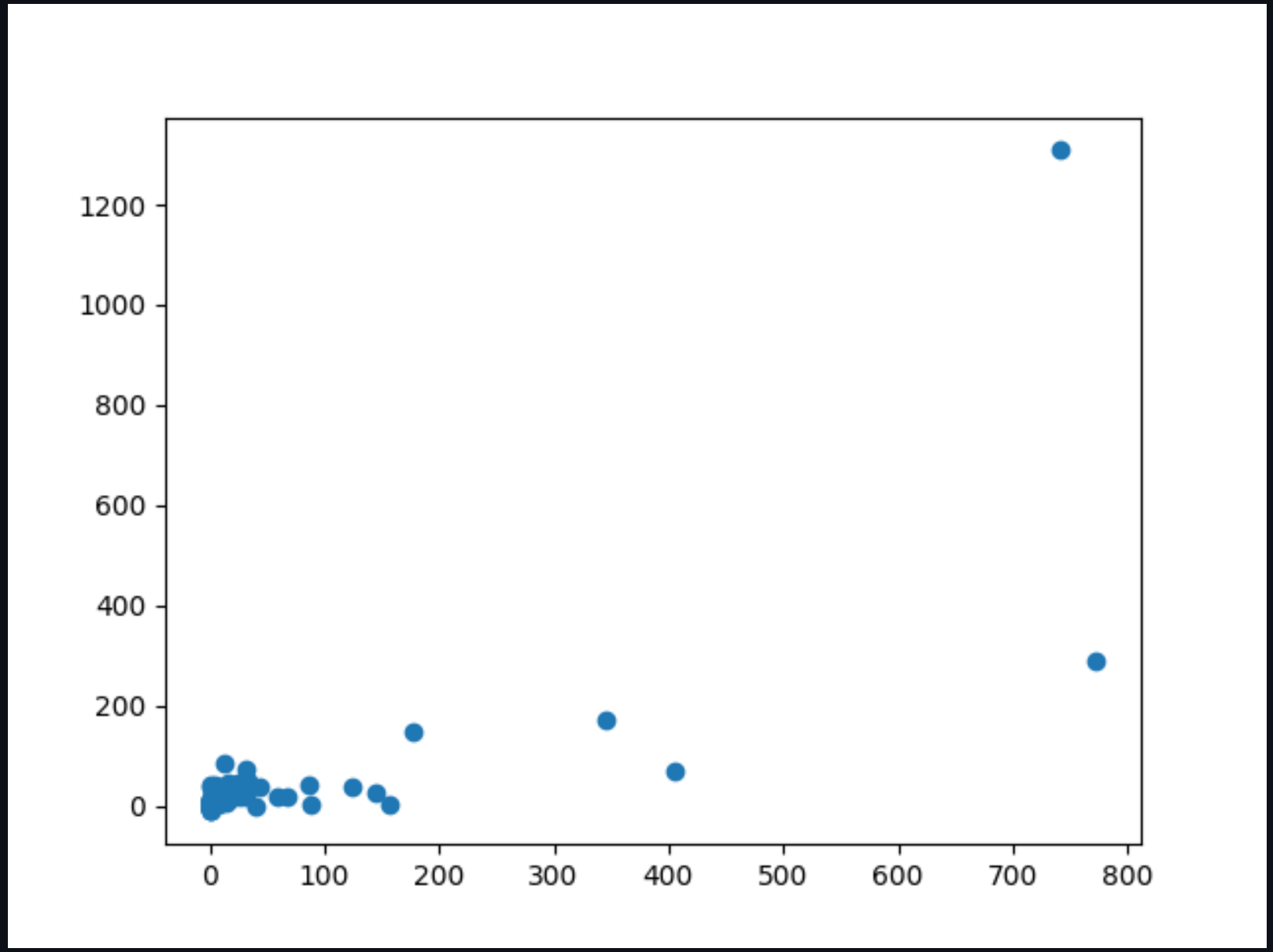
RepeatedStratifiedKFold - это вид кросс-валидации, который помогает учесть разнообразие данных и уменьшить вероятность переобучения модели. Его особенностью является стремление **сохранить баланс классов** в каждом фолде.

Модель LS

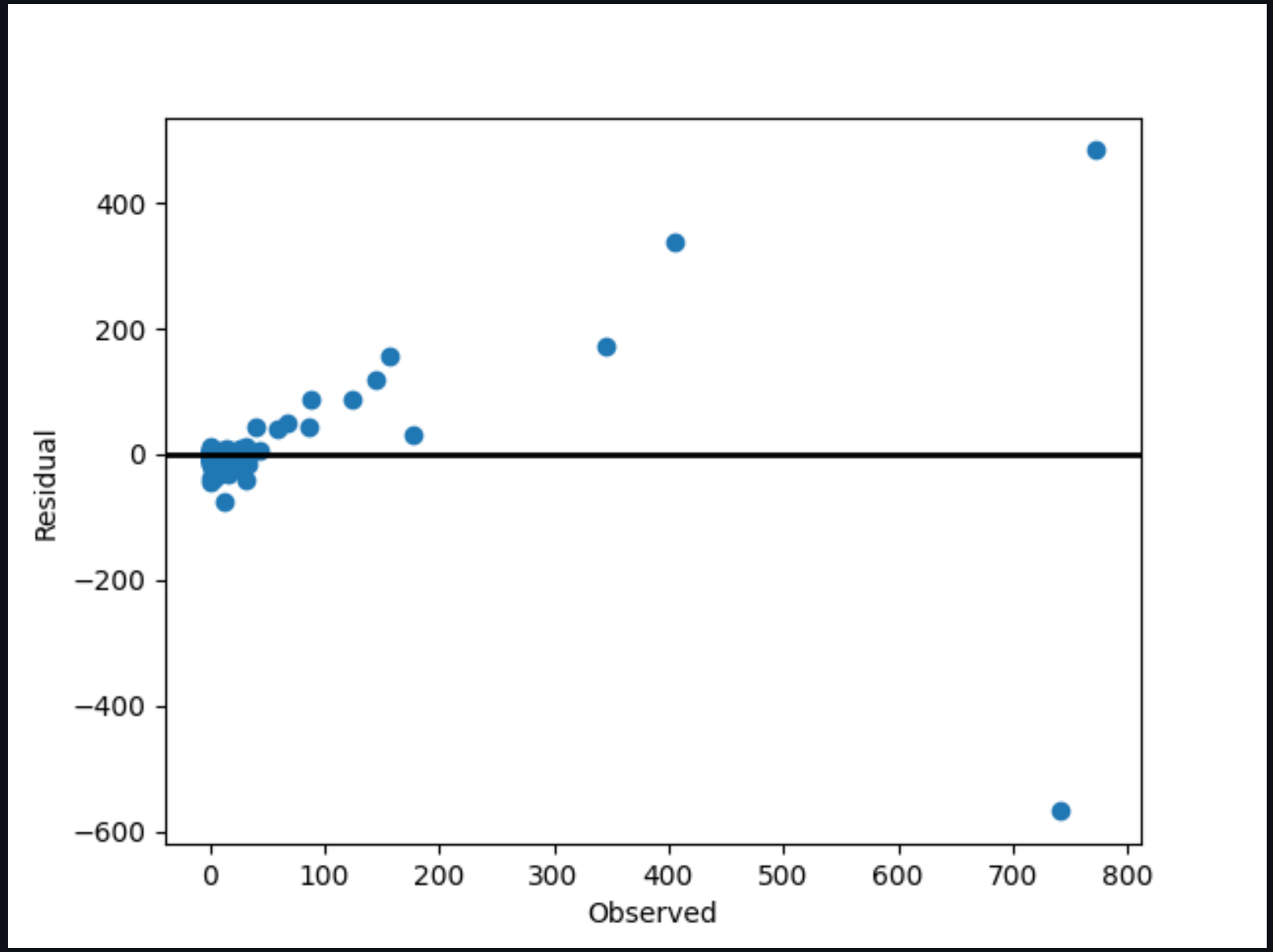
> Метод наименьших квадратов (Least squares)

LS заключается в поиске линейной функции, которая наилучшим образом соответствует данным путем минимизации суммы квадратов разницы между фактическими и предсказанными значениями. Метод **оптимизирует сумму квадратов остатков** и находит оптимальные значения коэффициентов линейной модели.

- **Дополнительный плюс** метода состоит в том, что он обеспечивает аналитические *(закрытые)* решения для оценки коэффициентов линейной модели
- **Но** довольно чуствителен к выбросам. Даже небольшие выбросы могут сильно исказить оценки коэффициентов регрессии и делать предсказания менее точными



Confusion plot for LS model



Confusion line plot for LS model

Результат работы моделей

- **R2 Score:** 0.4486
- **Среднее:** 30.8183
- **Дисперсия:** 7895.4442
- **СКО:** 88.8563

Подобранные параметры

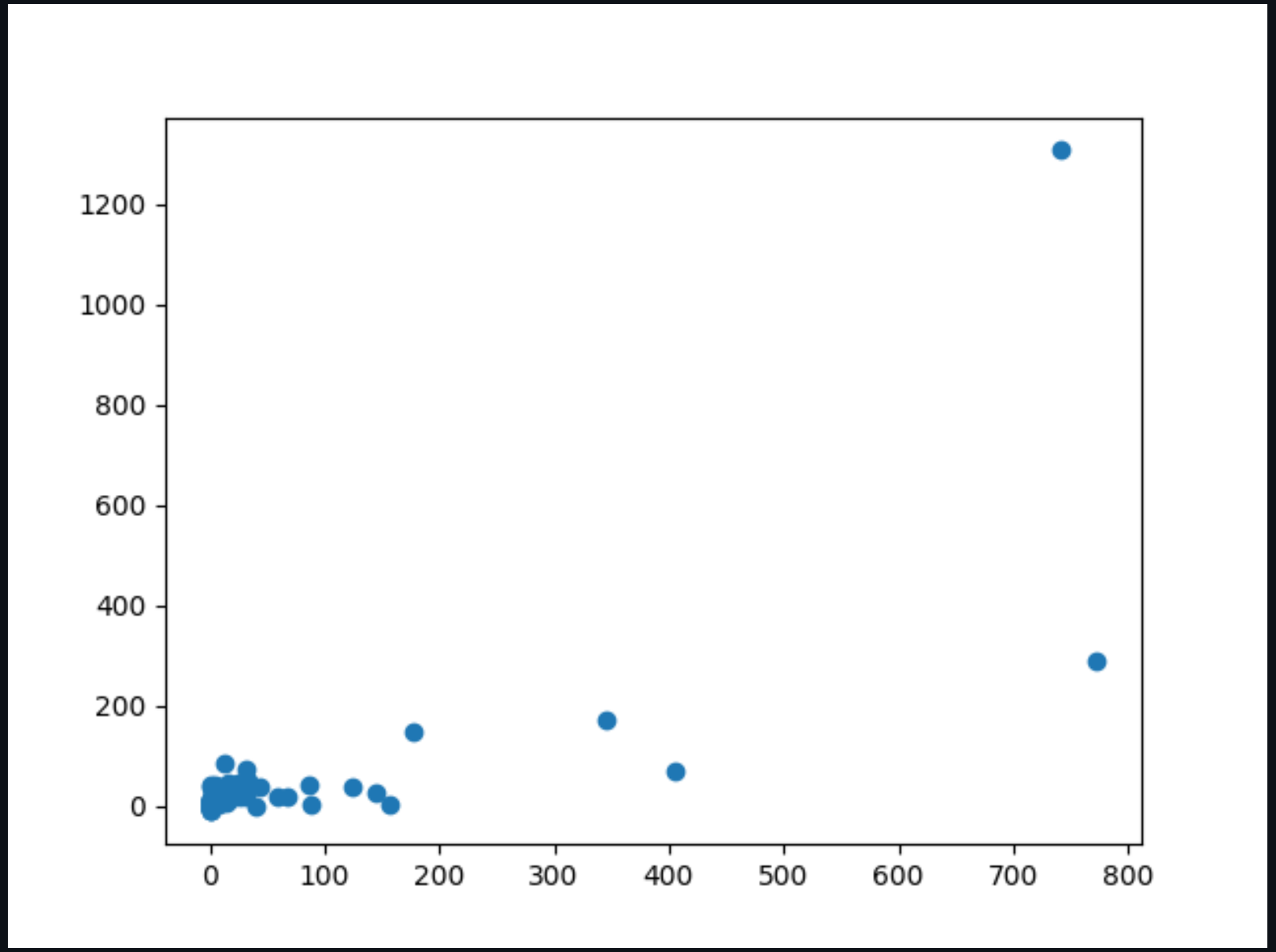
- copy_x: **True**
- fit_intercept: **False**

Модель Ridge

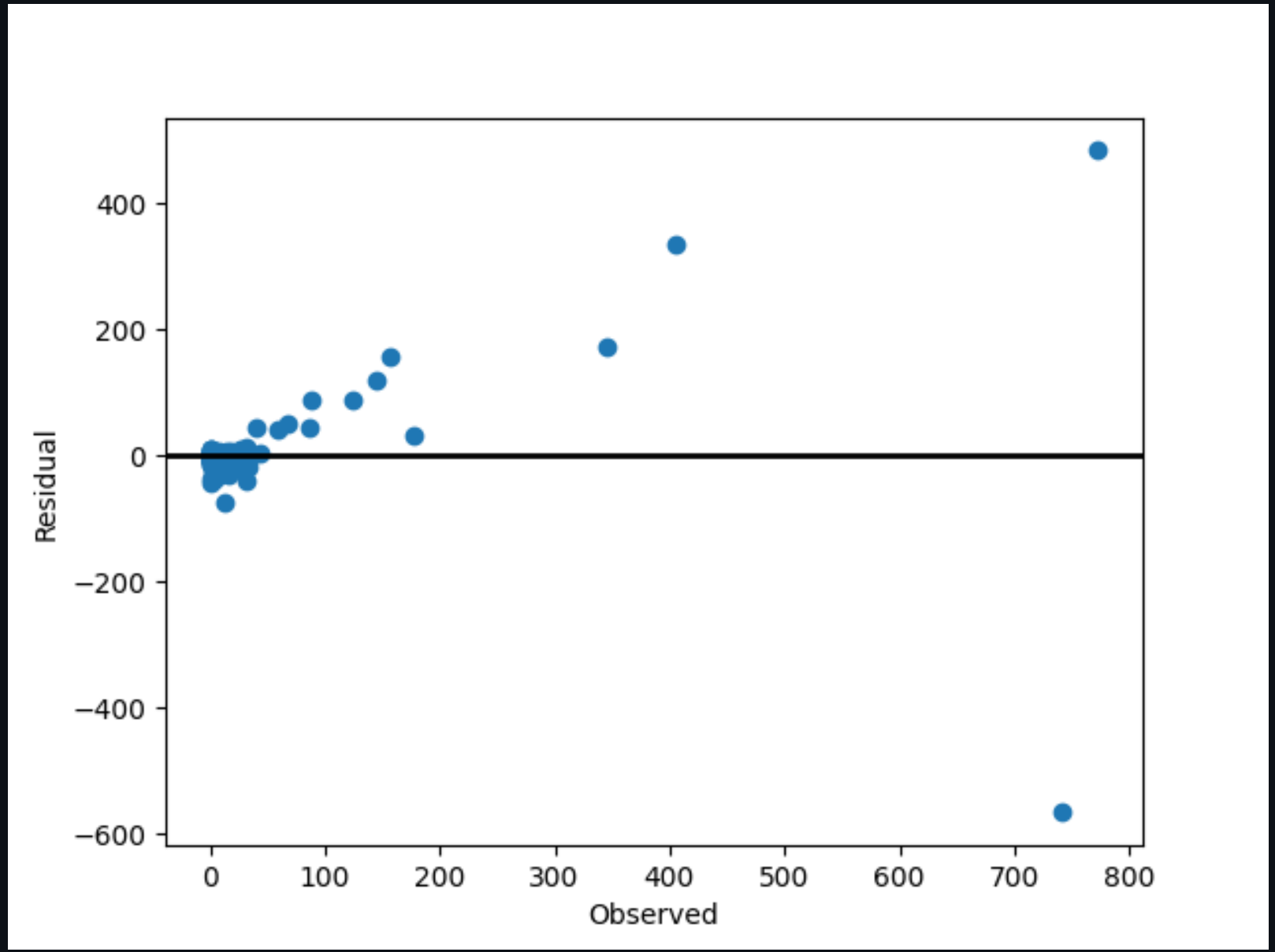
> Ridge ♪

Ridge очень похож на **LS**, ведь он также минимизирует сумму квадратов, но к этой сумме добавляется **штрафование больших значений коэффициентов модели**, что способствует снижению их величины и предотвращает переобучение.

- **Явным плюсом** является стабильность метода, ведь он менее чувствителен к выбросам
- **Но** Из-за регуляризации коэффициенты в Ridge регрессии могут быть менее интерпретируемыми, чем в обычной линейной регрессии, потому что они могут быть уменьшены или даже **занулены**



Confusion plot for Ridge model



Confusion line plot for Ridge model

Результат работы моделей

- **R2 Score:** 0.4497
- **Среднее:** 30.8025
- **Дисперсия:** 7880.2552
- **СКО:** 88.7708

Подобранные параметры

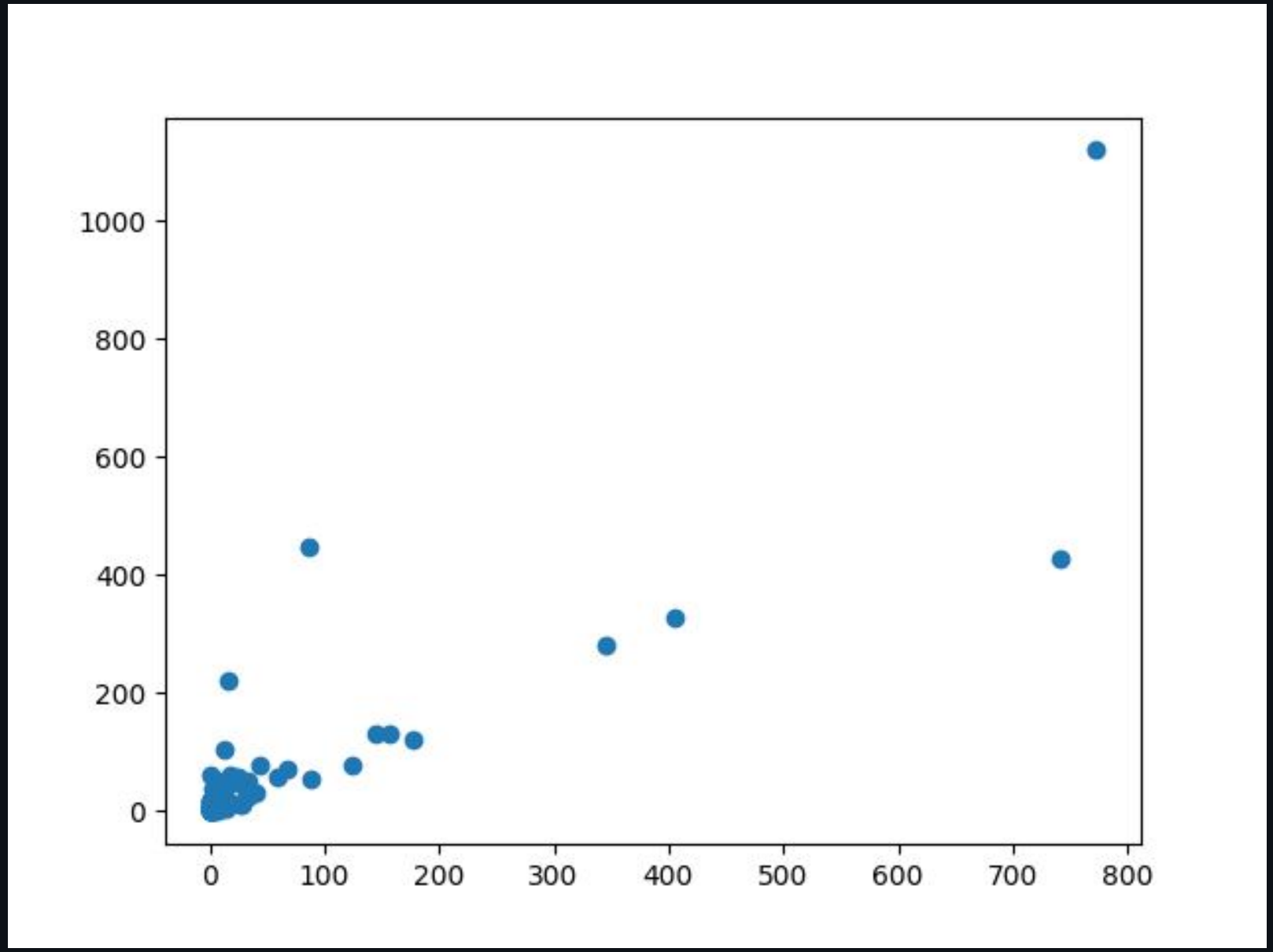
- Альфа: **100**
- Copy_x: **True**
- Fit_intercept: **True**
- Solver: **Cholesky**

Модель DT

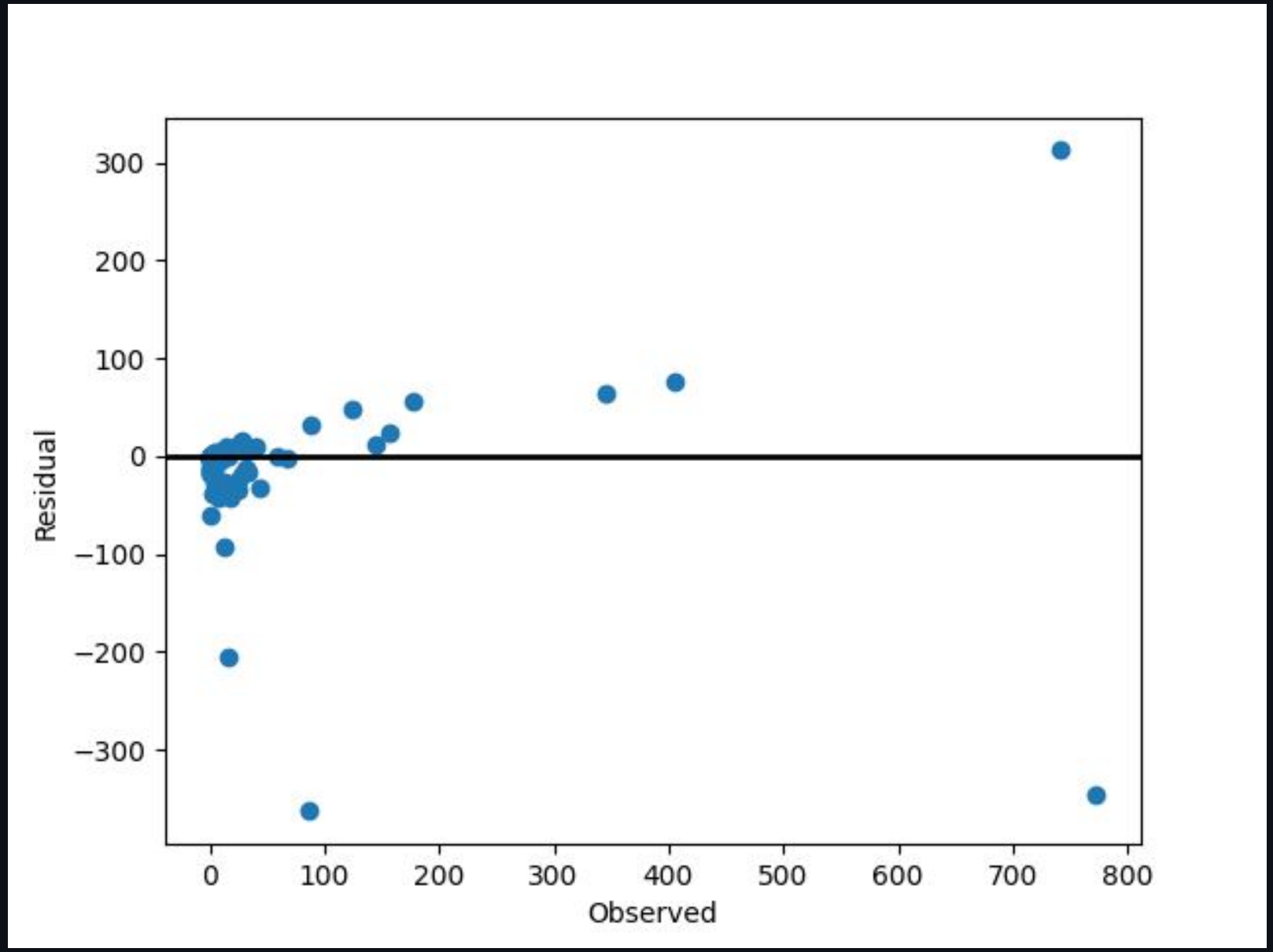
> Decision Tree (Регрессионное дерево)

DT - это метод регрессии, основанный на построении **дерева решений**, где каждый узел представляет собой тест по одному из признаков. Он разбивает данные на подгруппы на основе значений признаков и прогнозирует целевую переменную для каждой подгруппы

- Чем глубже дерево, тем сложнее правила принятия решений и тем лучше модель.
- Она **хорошо** справляется с обработкой **нелинейных отношений**
- **Но** слишком склонна к переобучению *(возможен случай: хорошо на training, плохо на test)*



Confusion plot for DT model



Confusion line plot for DT model

Результат работы модели

- **R2 Score:** 0.6934
- **Медиана:** 23.1270
- **Дисперсия:** 4349.8
- **СКО:** 65.953

Подобранные параметры

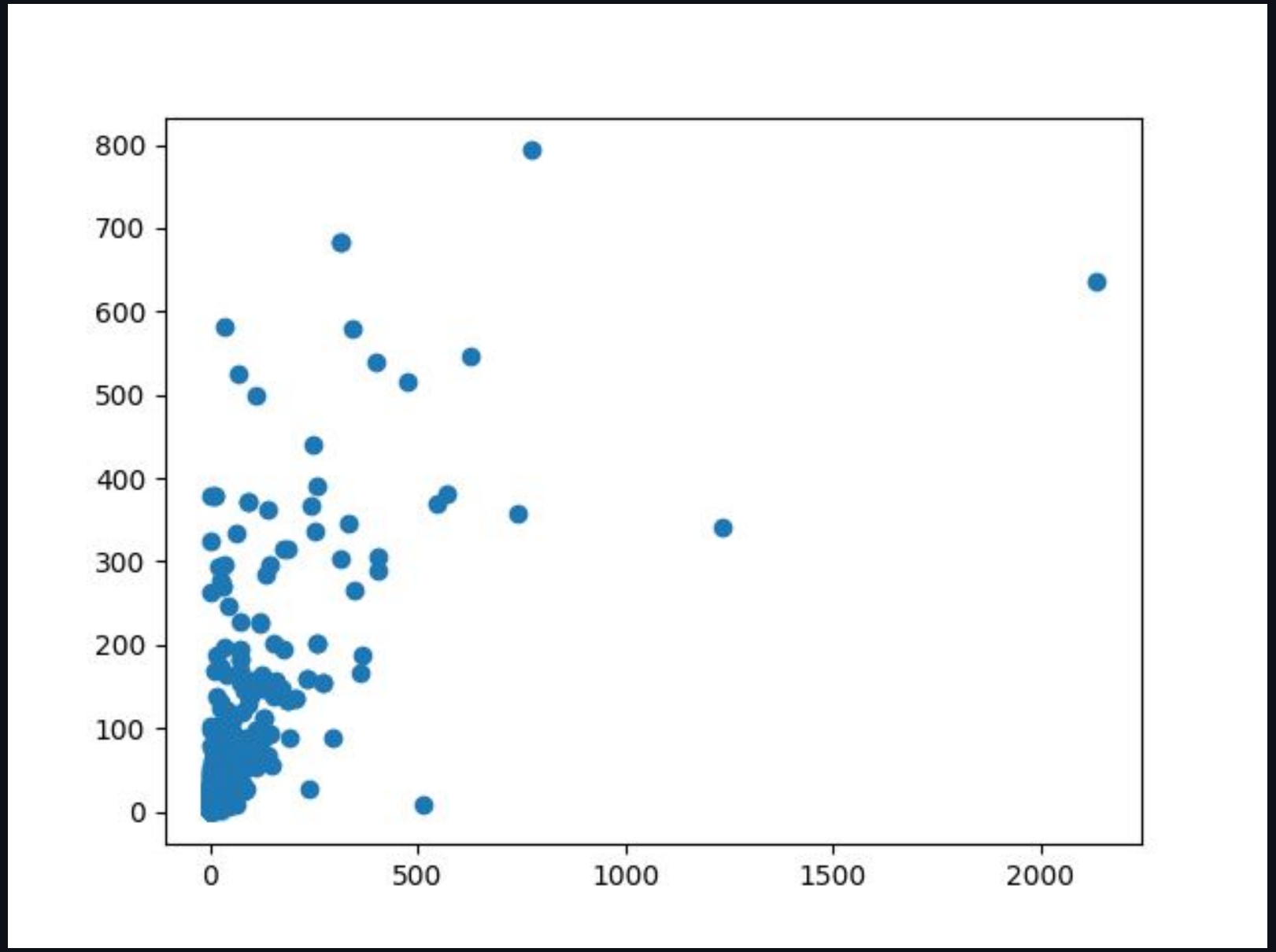
- Выбранный критерий: **squared_error**
- Глубина дерева: **10**
- Максимальное к-во листьев: **105**
- Минимальное к-во экземпляров в листе: **85**

Модель RF

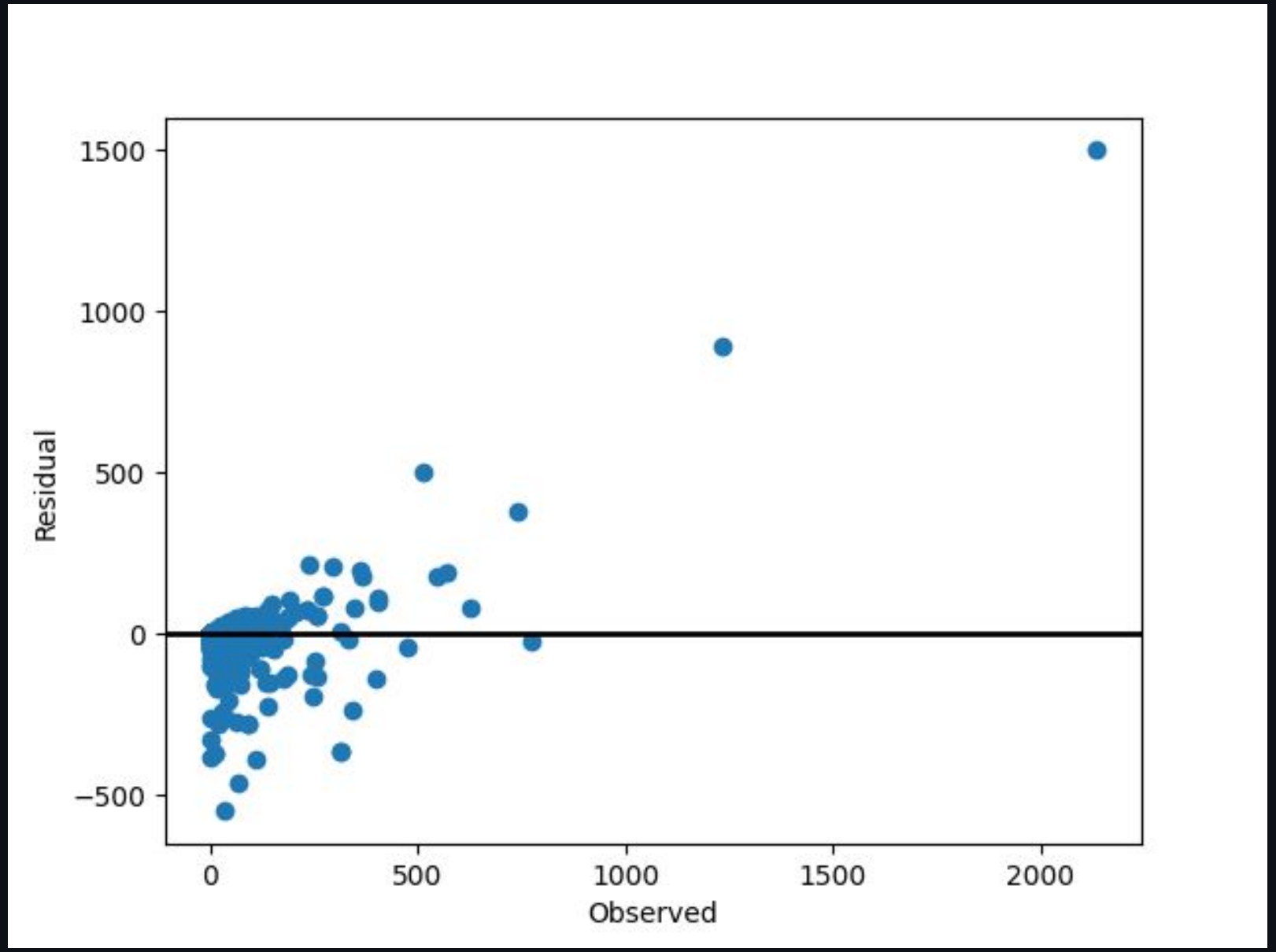
> Random Forest

RF продолжает идею **DT**, ведь он создает лес из этих деревьев. При предсказании модель **усредняет (или взвешивает) предсказания** всех деревьев, что позволяет уменьшить дисперсию и повысить точность предсказаний.

- **Огромный плюс** - высокая точность модели. ~~Верьте на слово~~
- **Но также весомый минус** - из-за большого количества деревьев, модель может быть очень вычислительно затратна



Confusion plot for RF model



Confusion line plot for RF model

Результат работы модели

- **R2 Score:** 0.7651
- **Медиана:** 23.0837
- **Дисперсия:** 3332.65
- **СКО:** 57.729

Подобранные параметры

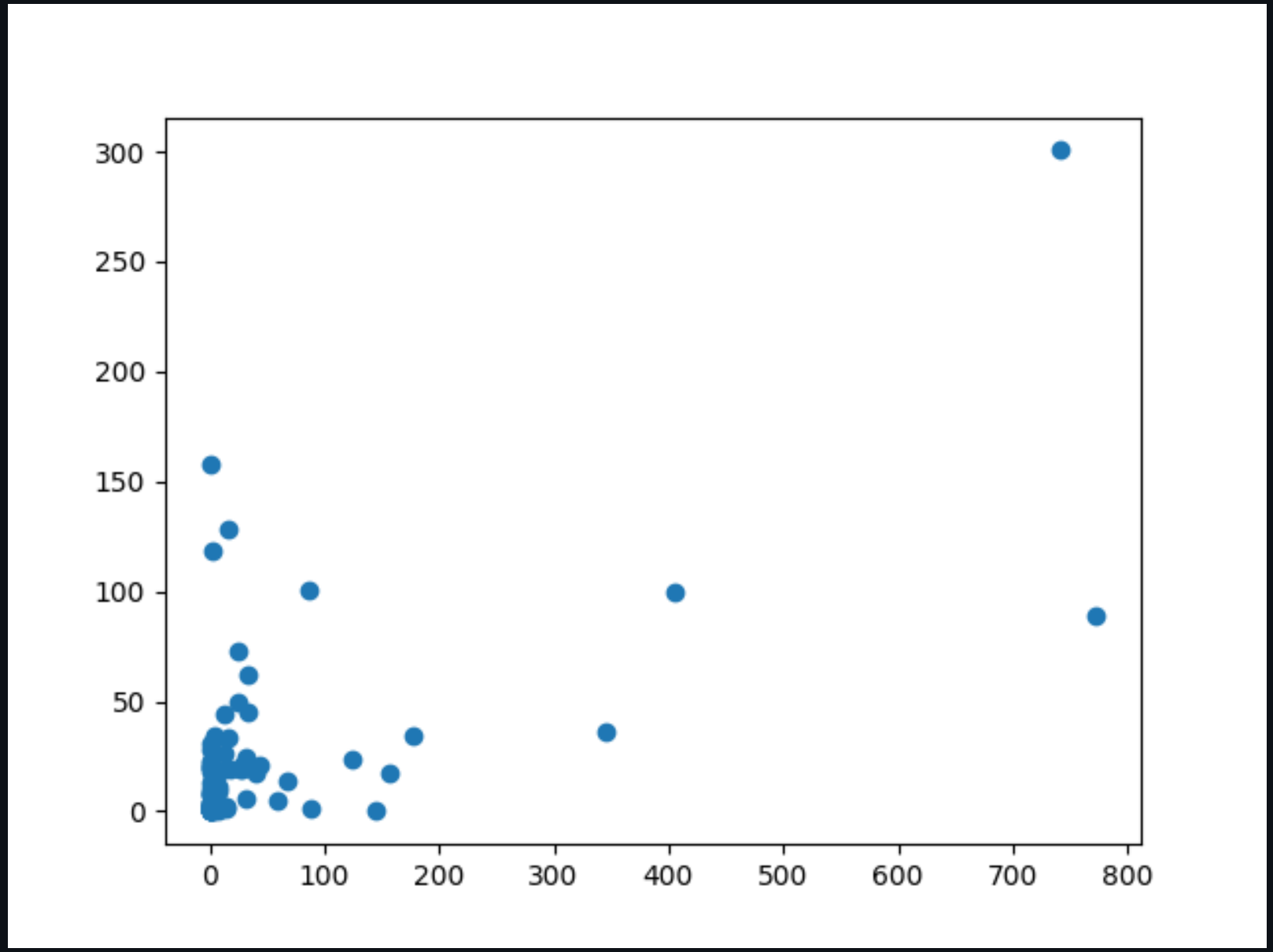
- Глубина дерева: **15**
- Максимальное к-во листьев: **105**
- Минимальное к-во экземпляров в листе: **55**
- К-во эстиматоров: **50**

Модель KNN

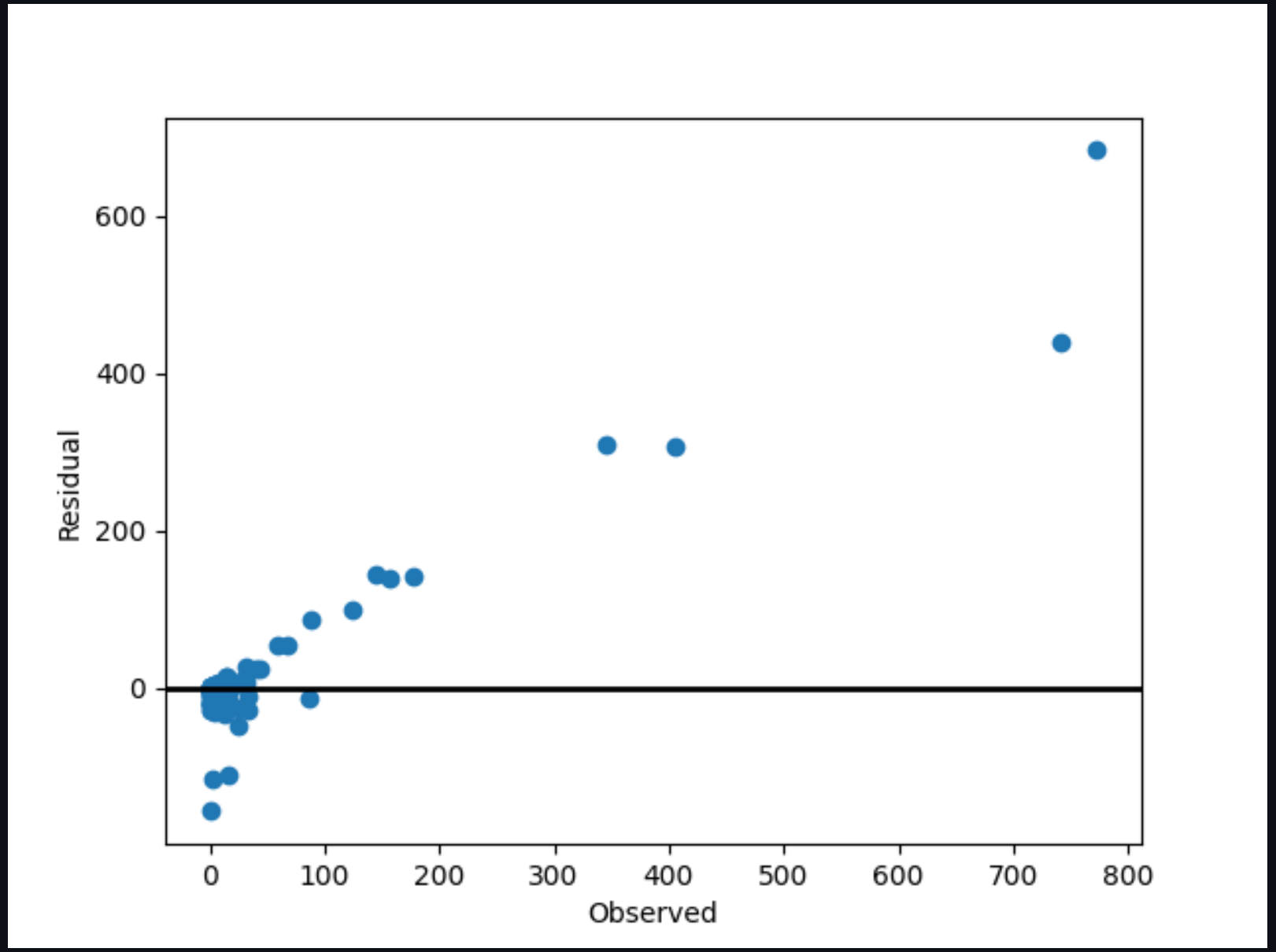
> K Nearest Neighbors (Регрессор "k-ближайших соседей")

Суть **KNN** заключается в том, что он определяет прогноз для нового объекта на основе **среднего** *(или медианного)* значения целевой переменной у её **ближайших соседей** в обучающей выборке.

- Модель **хороша**, поскольку ее метод не использует сложную математику, а реализация проста и очевидна.
- **Но не без проблем**: модель очень вычислительно затратна, особенно на больших наборах данных.



Confusion line for KNN model



Confusion line plot for KNN model

Результат работы модели

- **R2 Score:** 0.2941
- **Медиана:** 36.2177
- **Дисперсия:** 10107.9798
- **СКО:** 100.5384

Подобранные параметры

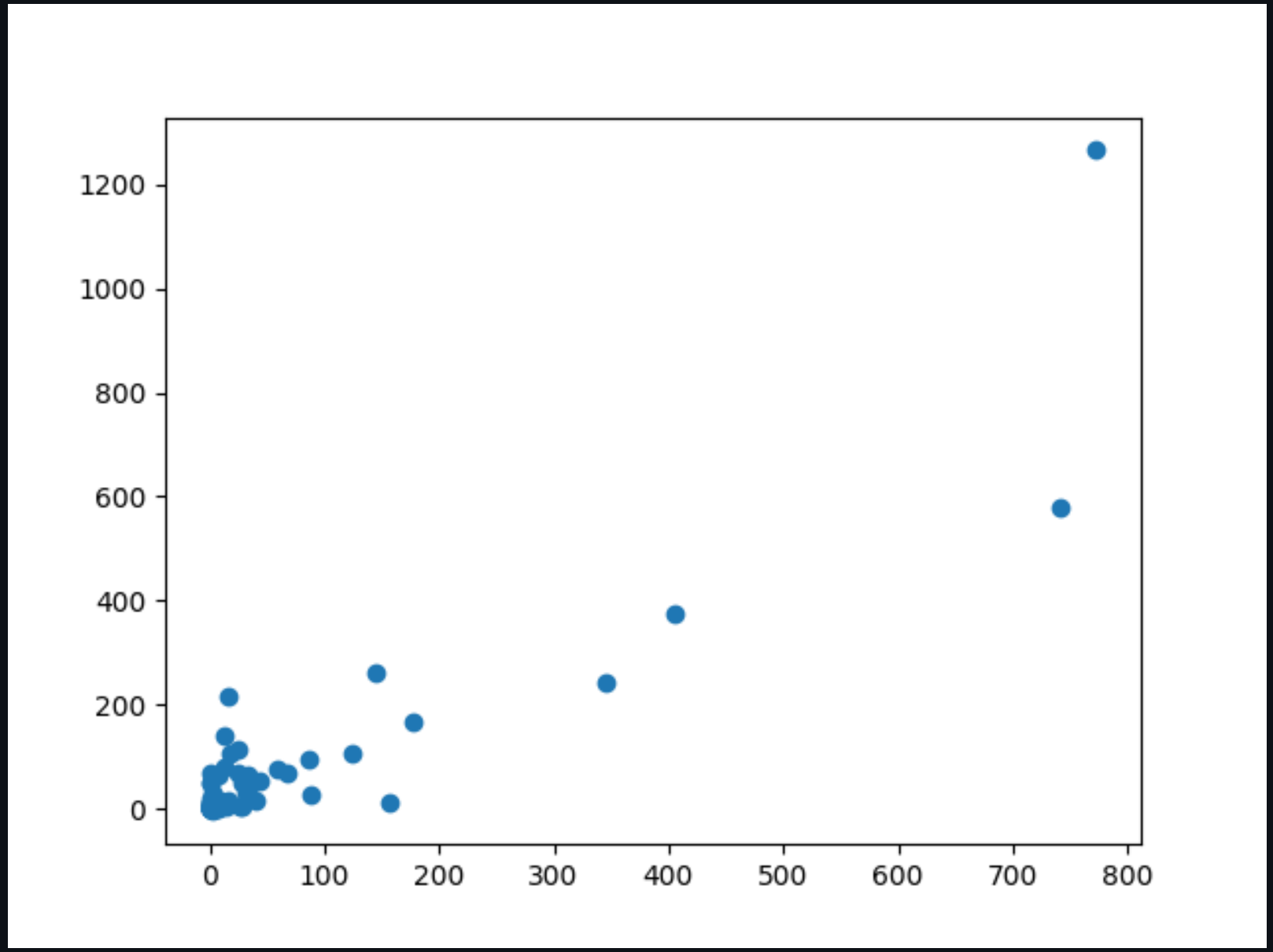
- Алгоритм: **kd_tree** *(p = 1)*
- К-во листов: **15**
- К-во соседей: **6**
- Параметр веса: **Расстояние**

Модель GB

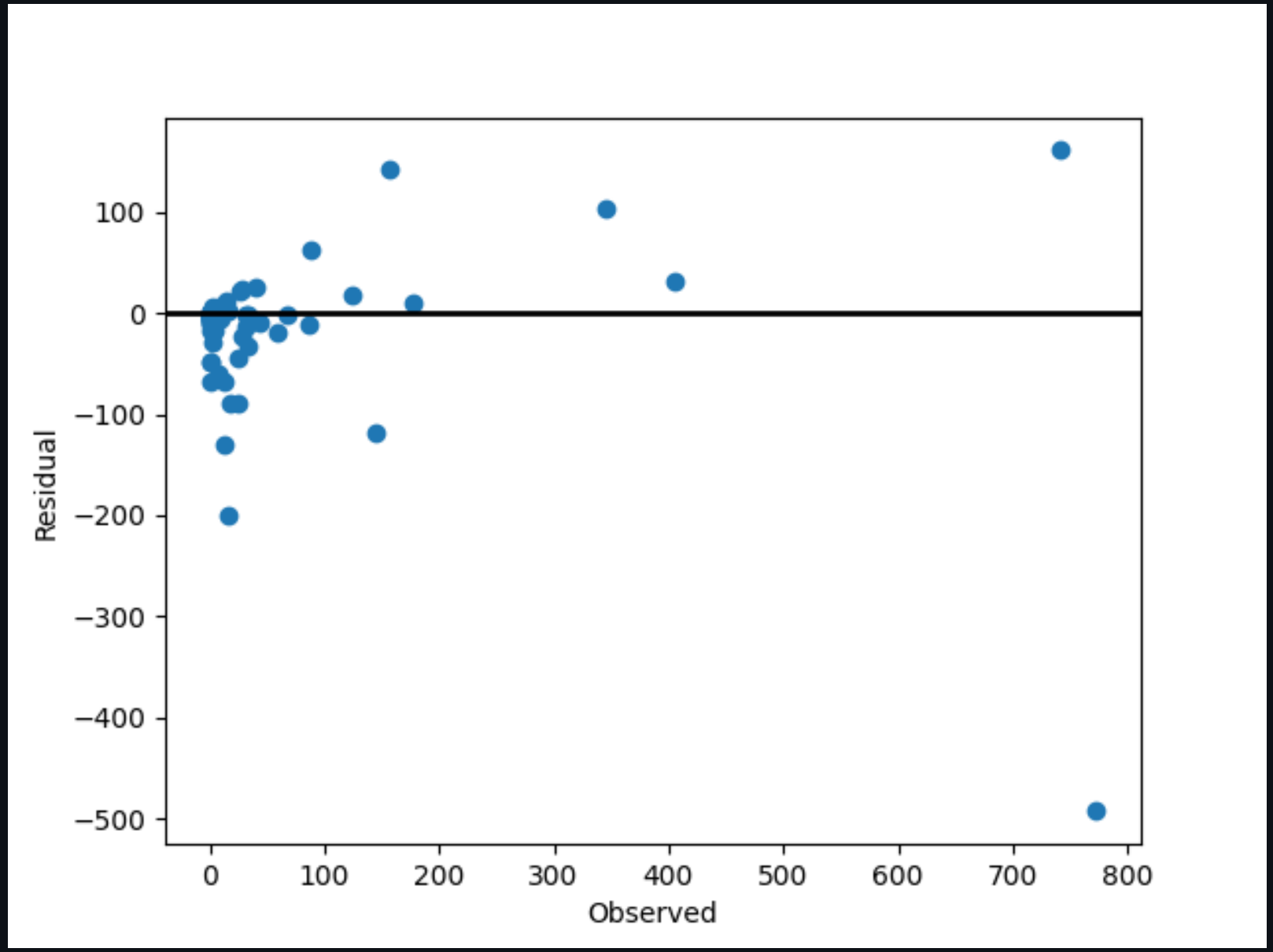
> Gradient Boosting

GB работает путем последовательного добавления новых **моделей (деревьев)** к существующему ансамблю. Каждое новое дерево обучается на данных, учитывая ошибки, сделанные предыдущими деревьями.

- **Плюсом метода** является его способность автоматически учитывать важность признаков.
- **Но к сожалению,** она вычислительно затратна и если недостаточно ограничить **глубину деревьев или количество итераций**, модель может переобучиться на обучающих данных, что приведет к плохой обобщающей способности модели.



Confusion plot for GB model



Confusion line plot for GB model

Результат работы модели

- **R2 Score:** 0.7049
- **Медиана:** 24.4965
- **Дисперсия:** 4225.0200
- **СКО:** 65.0001

Подобранные параметры

- Глубина: **6**
- Скорость обучения: **1**
- random_state: **42**
- n_estimators: **3**

Итоги

В ходе проделанной работы было выявлено, что лучшим регрессором для поиска целевой переменной оказалась модель **Random Forest** со значением **R2 = 0.7651** 🥳🥳🥳

Почему так?

- В данных малозаметны линейные зависимости
- Деревья хороши, а много деревьев лучше *(целый лес!)*
- Так опять вышло 😄😄😄

Работали,

Data Balbesing ♂️

> это мы ~~пытаемся добавить гифку в пдф~~ следим за выступлениями остальных групп ->