

Доклад по практике анализа данных

Прогнозирование цены лего-набора с помощью методов регрессии 🧱 🌟

Работу подготовил: Бабушкин Михаил КЭ-301

Постановка задачи ❄️❄️❄️

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

Необходимо **реализовать несколько моделей** и полученные результаты всех методов **сравнить** между собой.

Выберем лучшую 🤖 из худших 🤖 полученных моделей и покажем полученные результаты.

Информация о датасете

- Дан датасет со всеми наборами лего с 1955 по 2023 год
- Размер: 1,12ГБ 😁

Этот набор данных содержит наборы LEGO, взятые с сайта lego.com. Каждое наблюдение представляет собой отдельный набор LEGO, и есть такие характеристики, как количество деталей в наборе, стоимость набора и т.д. Этот набор данных содержит наборы LEGO из всех разных стран, в которых они продаются онлайн.

Lego sets and price [1955 - 2023]

Unified Rebrickable database with price and star ratings added informations



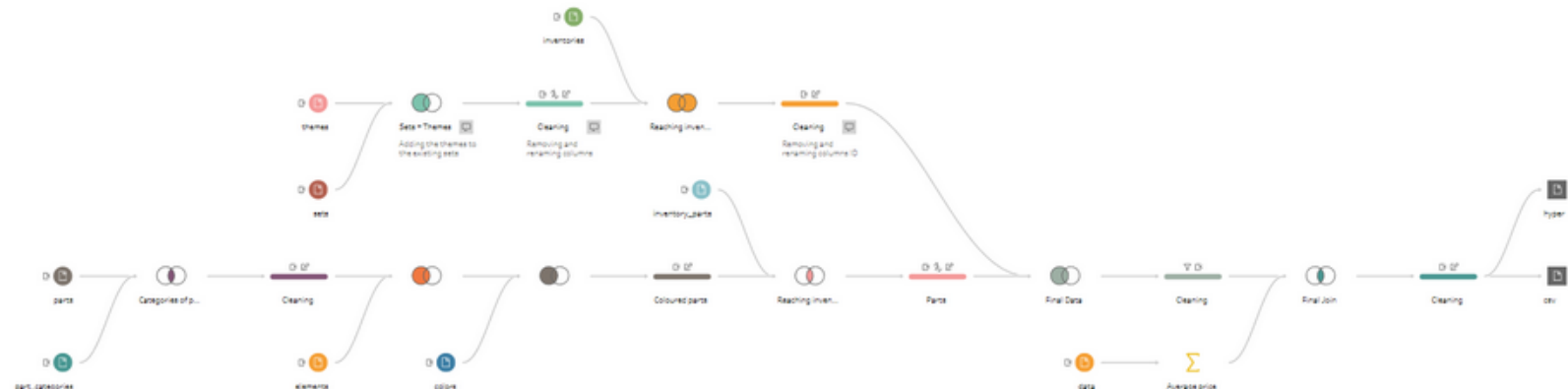
[Data Card](#) [Code \(2\)](#) [Discussion \(0\)](#)

About Dataset

This dataset has been created by merging two types of dataset:

- [rebrickable dataset](#) downloaded on 2023.11.14
- informations about price, reviews and star rating for each set coming from [this](#) dataset

Preprocessing steps have been computer with Tableau Prep (the whole steps are represented in the image below), and involved cleaning, joining and aggregating data.



Usability ⓘ

10.00

License

[CC0: Public Domain](#)

Expected update frequency

Never

Tags

Tabular

Classification

Regression

Clustering

Bigquery

14 переменных

Переменная	Описание
year	год создания наборов
Theme name	тема, к которой относится набор
Sets name	название набора
Sets URL	URL изображения набора
Part category	категория детали
Part name	название каждой детали, входящей в набор
Part material	материал детали

Целевая переменная - 'Set Price'

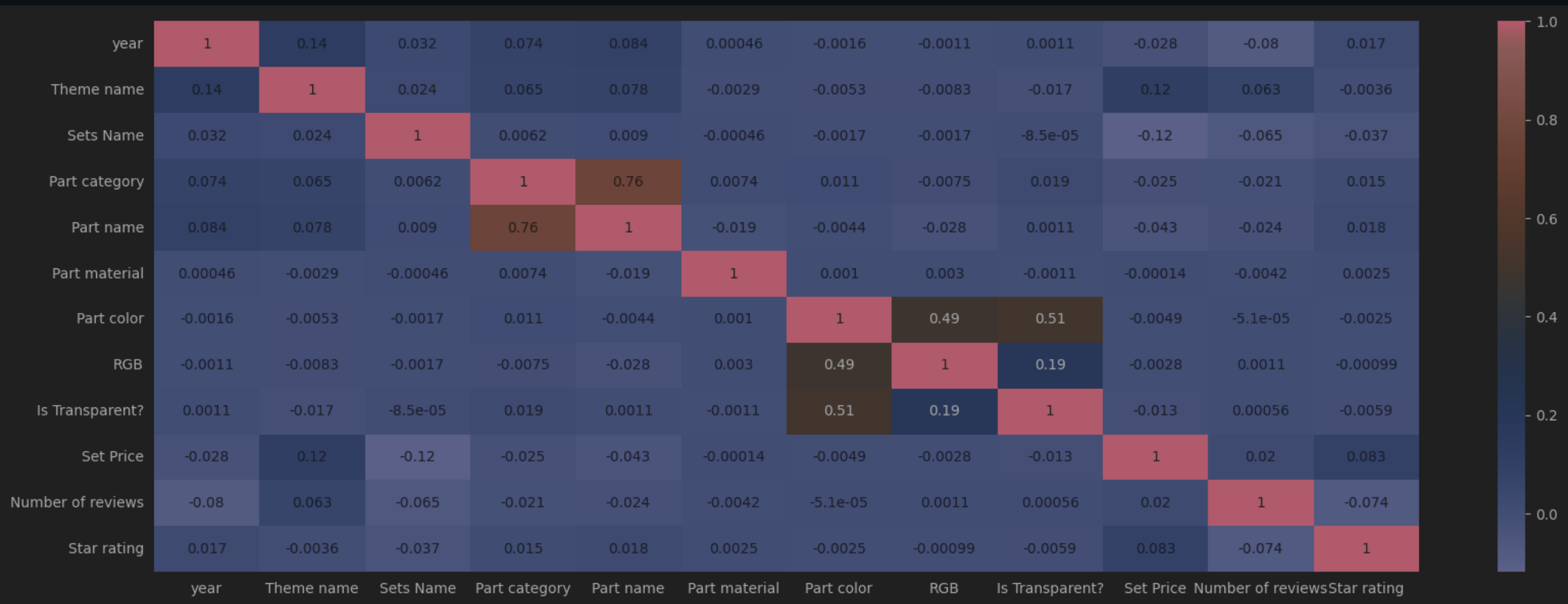
Переменная	Описание
Part color	цвет детали
RGB	RGB цвет детали
Is Transparent	прозрачна ли деталь или нет
Part URL	URL-изображение для каждой части
Set Price	средняя цена за каждый комплект
Number of reviews	среднее количество отзывов для каждого набора
Star rating	средний рейтинг на сайте лего для каждого набора

Тепловая карта

Тепловая карта - это визуальный инструмент, используемый в разведочном анализе данных для отображения матрицы данных с помощью цветовой шкалы. Зачастую тепловая карта представляет собой двумерную таблицу, где каждая ячейка содержит числовое значение, преобразованное в цвет.

Ее использование может быть полезным для:

1. **Визуализации** корреляции
2. **Выявления** аномалий
3. **Отслеживания** распределения
4. **Информационного обобщения**



Объяснение

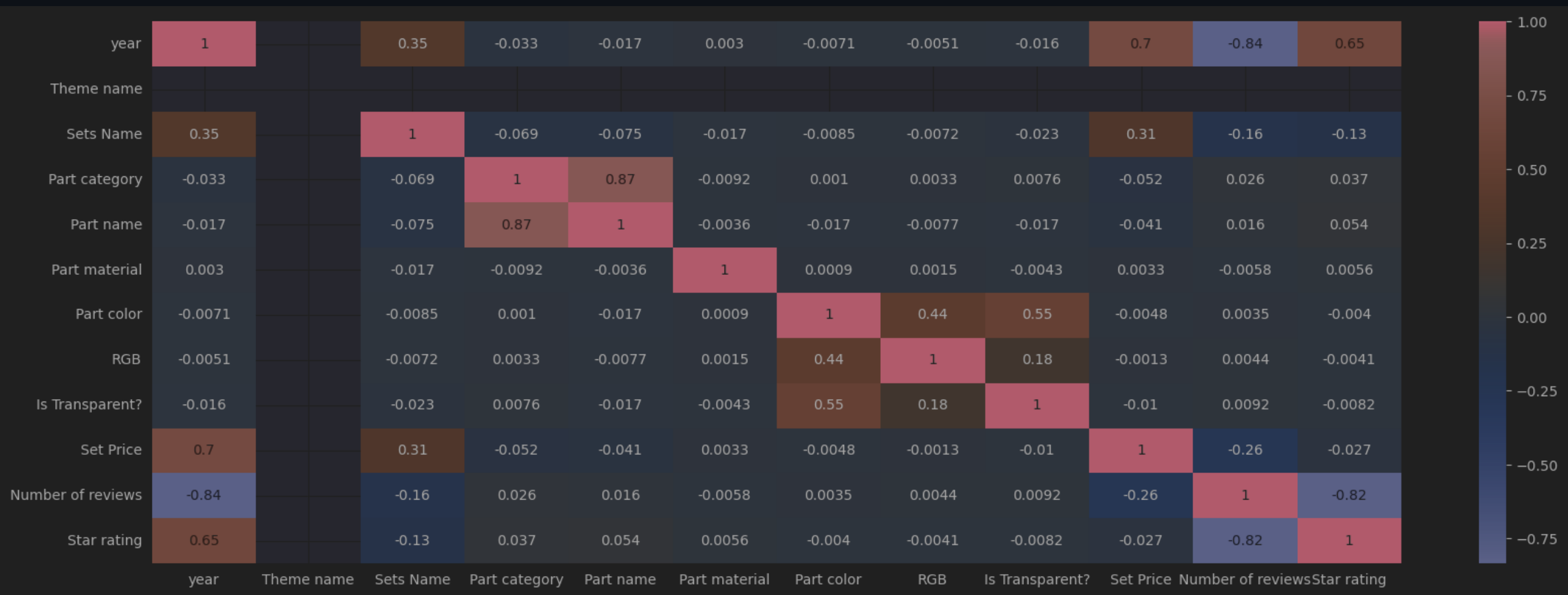
Корреляция заметна, но не для 'Set Price'

Что делать?

Предположение: цена LEGO набора зависит от той 'вселенной' по референсу на которую его выпускают

И?

Сделаем селектор!



Теперь корреляция лучше!


- 'Set Price' & 'year' -> 0.7
- 'Set Price' & 'name' -> 0.31

Что дальше?

Пишем свой селектор

Селектор

- Это инструмент для выбора определенного подмножества данных из более крупного набора, часто используемый для фильтрации или извлечения конкретных столбцов, строк или признаков в зависимости от заданных критериев или условий.
- В отличие от ансамблирования, где на итог отдается взвешенное решение множества моделей, селектор выбирает из множества признаков один наиболее выделяющийся и разбивает весь датасет так, чтобы каждому уникальному значению этого признака ассоциировалось подмножество данных, где находится это значение.
- Был реализован класс **SelectiveWrapper** представляющий собой "обертку" для обучения модели по **selected variable**.

```
10 class SelectiveWrapper(RegressionModelApi):
11     # Selective variable
12     __selective_variable_name: str
13
14     # flag is trained
15     __is_trained: bool
16
17     # Solver models
18     __solvers: Dict[Any, RegressionModelApi]
19
20     # Solver model factory
21     __factory: Callable[[], RegressionModelApi]
22
23     # Overall report (used to describe model state after training & testing)
24     __report: Dict[str, Any]
25
26      AlexCawl
27     def __init__(self, selector_name: str, model: Callable[[], RegressionModelApi]):
28         self.__selective_variable_name = selector_name
29         self.__is_trained = False
30         self.__factory = model
31         self.__solvers = dict()
32         self.__report = dict()
33         self.__report.update(
34             {
35                 "SELECTOR_NAME": self.__selective_variable_name,
36                 "SELECTOR_MODEL": self.__factory().__class__.__name__
37             }
38         )
```

Выбор моделей

Для решения проблемы регрессии я решил выбрать следующие модели, и распределить их между собой.

- LS, Ridge - Михаил Бабушкин
- RF - Михаил Дедушкин
- GB - Бабушкин М.В.
- CV - Сервер на Linux

Почему нет нейронки?



У меня нет вычислительных
мощностей для нейронки на
1,2ГБ сырых данных...

сервер умер 🤡 ->

```
mick@RECHNER: ~/RemoteProjects/LegoPriceResearchLab

b/runner/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py", line 1612, in _score
b/runner/lib/python3.10/site-packages/sklearn/utils/_param_validation.py", line 187, in wrapper
b/runner/lib/python3.10/site-packages/sklearn/metrics/_regression.py", line 989, in r2_score
targets(
b/runner/lib/python3.10/site-packages/sklearn/metrics/_regression.py", line 101, in _check_reg_targets
pe=dtype)
b/runner/lib/python3.10/site-packages/sklearn/utils/validation.py", line 957, in check_array
b/runner/lib/python3.10/site-packages/sklearn/utils/validation.py", line 122, in _assert_all_finite
b/runner/lib/python3.10/site-packages/sklearn/utils/validation.py", line 171, in _assert_all_finite_element_wise

rning)
/lib/python3.10/site-packages/sklearn/model_selection/_search.py:979: UserWarning: One or more of the test scores are non-finite: [
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
onds
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
/lib/python3.10/site-packages/sklearn/neural_network/_base.py:173: RuntimeWarning: overflow encountered in square
/lib/python3.10/site-packages/sklearn/neural_network/_multilayer_perceptron.py:1625: DataConversionWarning: A column-vector y was passed as a 1D array, which has been converted to 2D array by the error checking code. This conversion may not be correct,
).
/lib/python3.10/site-packages/sklearn/utils/extmath.py:192: RuntimeWarning: overflow encountered in matmul
/lib/python3.10/site-packages/numpy/core/_methods.py:118: RuntimeWarning: overflow encountered in reduce
e=where)
/lib/python3.10/site-packages/sklearn/neural_network/_base.py:173: RuntimeWarning: overflow encountered in square
/lib/python3.10/site-packages/numpy/core/_methods.py:118: RuntimeWarning: overflow encountered in reduce
e=where)
/lib/python3.10/site-packages/sklearn/utils/extmath.py:192: RuntimeWarning: overflow encountered in matmul
/lib/python3.10/site-packages/sklearn/utils/extmath.py:192: RuntimeWarning: invalid value encountered in matmul
/lib/python3.10/site-packages/sklearn/neural_network/_base.py:173: RuntimeWarning: overflow encountered in square
/lib/python3.10/site-packages/sklearn/utils/extmath.py:192: RuntimeWarning: overflow encountered in matmul
/lib/python3.10/site-packages/sklearn/utils/extmath.py:192: RuntimeWarning: invalid value encountered in matmul
/lib/python3.10/site-packages/sklearn/utils/extmath.py:192: RuntimeWarning: invalid value encountered in matmul
```

Рецепт настройки моделей от нашей команды

- **GridSearchCV**

Метод поиска по сетке находит наилучшую комбинацию параметров, которые дают наименьшую ошибку, путем обычного перебора: он создает модель для каждой возможной комбинации параметров.

- **Cross-Validation**

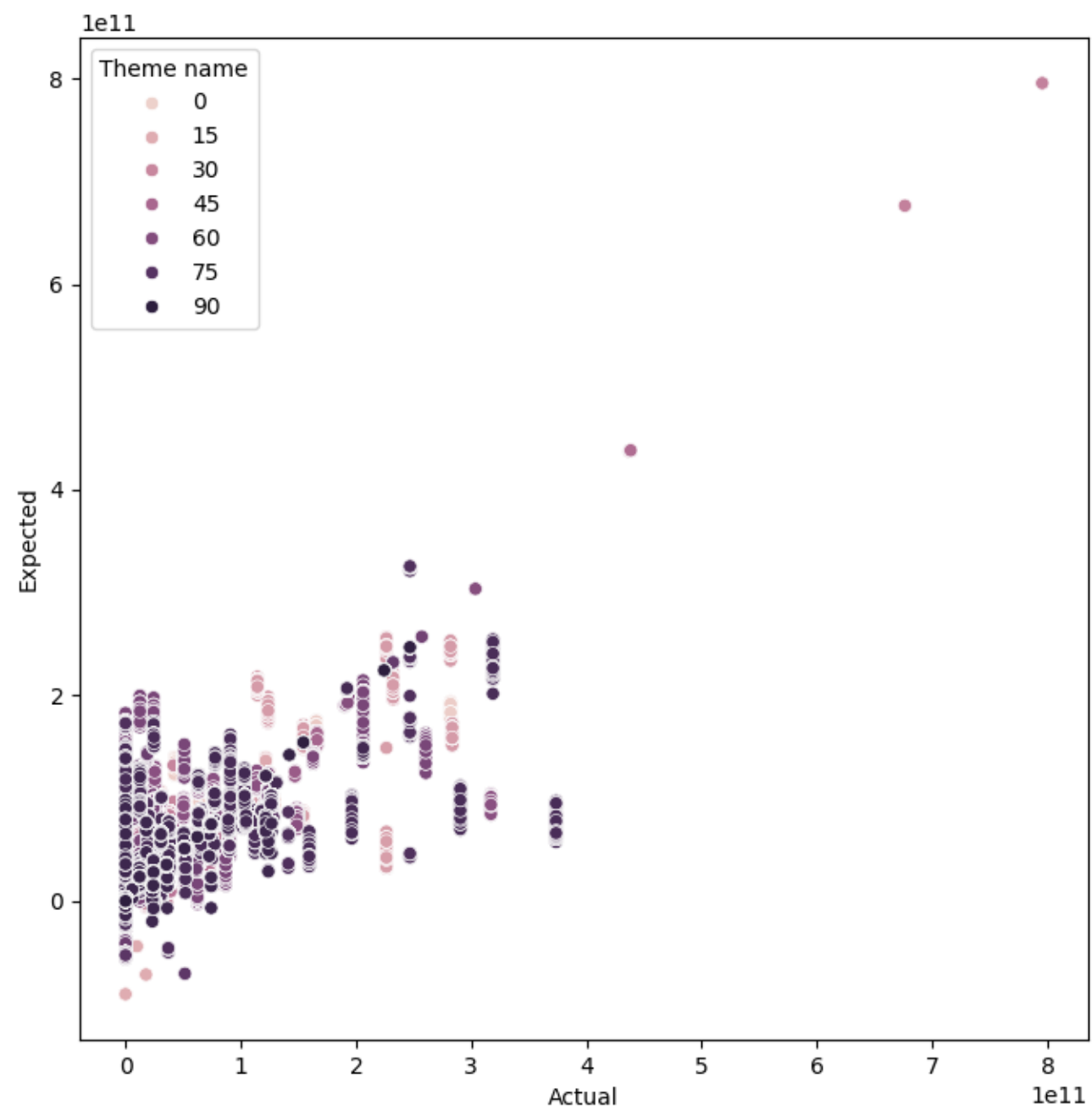
RepeatedStratifiedKFold - это вид кросс-валидации, который помогает учесть разнообразие данных и уменьшить вероятность переобучения модели. Его особенностью является стремление сохранить баланс классов в каждом фолде.

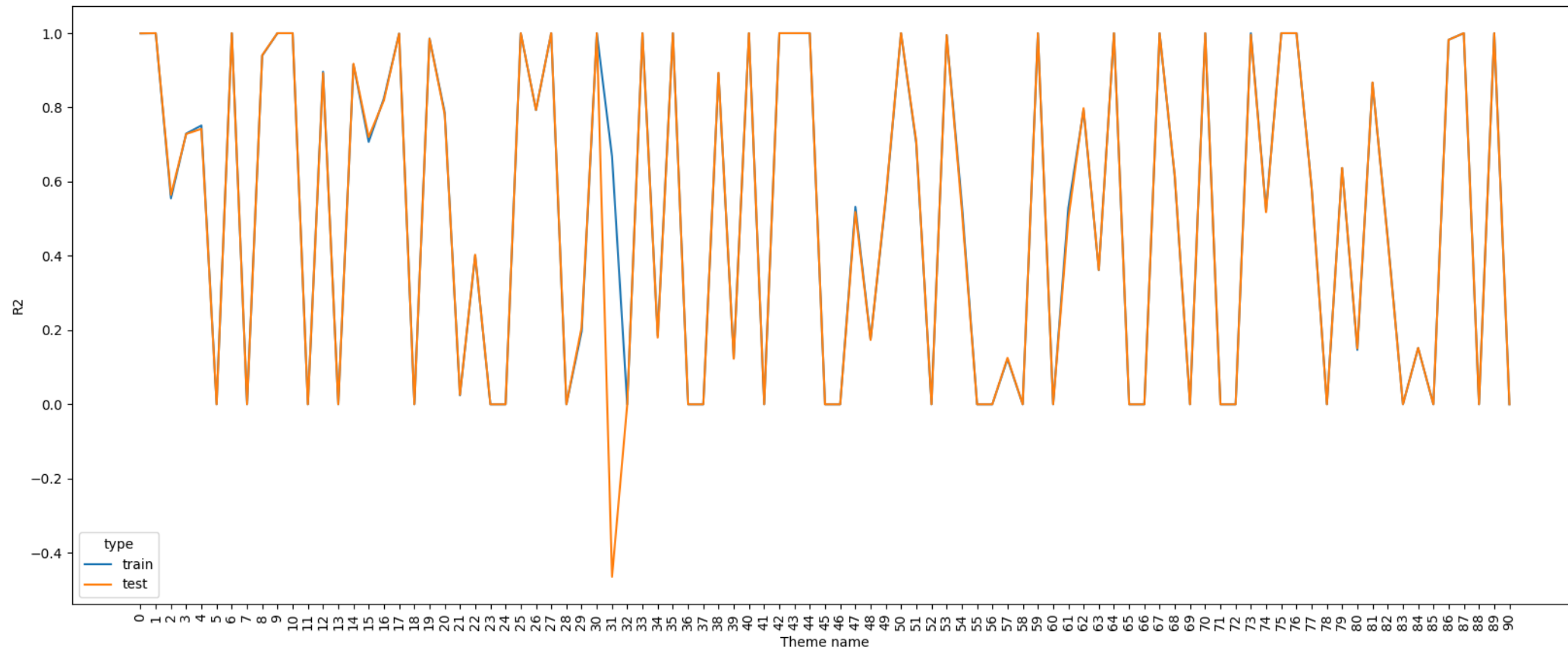
Модель LS

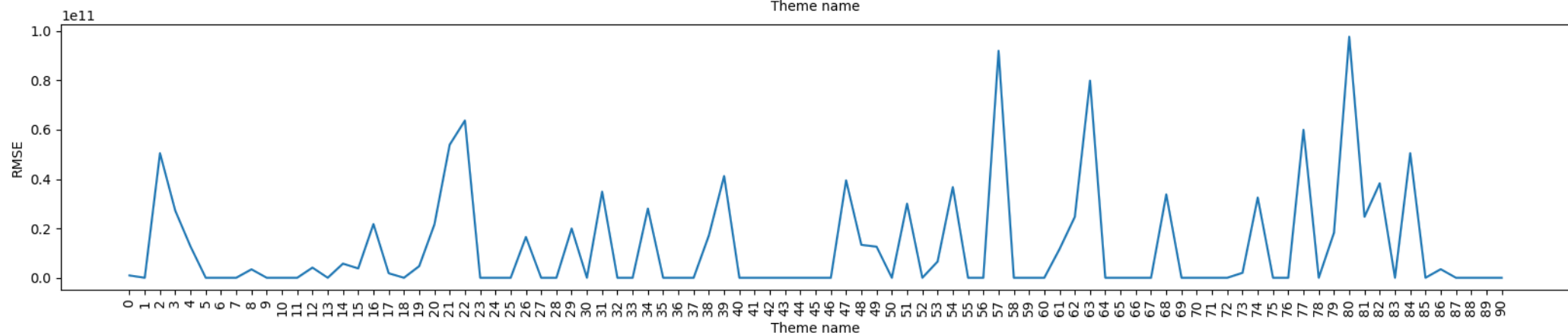
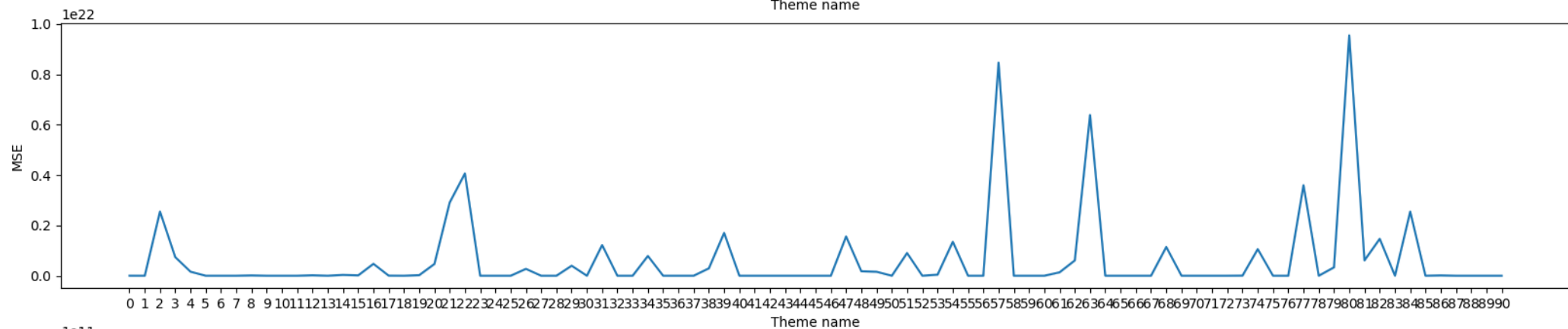
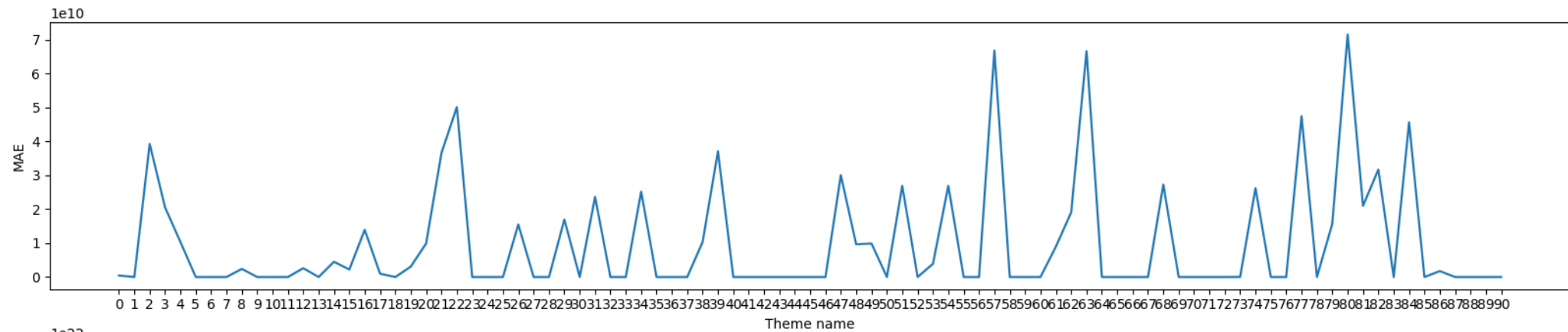
Метод наименьших квадратов (Least squares) 

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 (замкнутые) решения для оценки коэффициентов линейной модели.
- **Но** довольно чувствителен к выбросам. Даже небольшие выбросы могут сильно исказить оценки коэффициентов регрессии и делать предсказания менее точными.





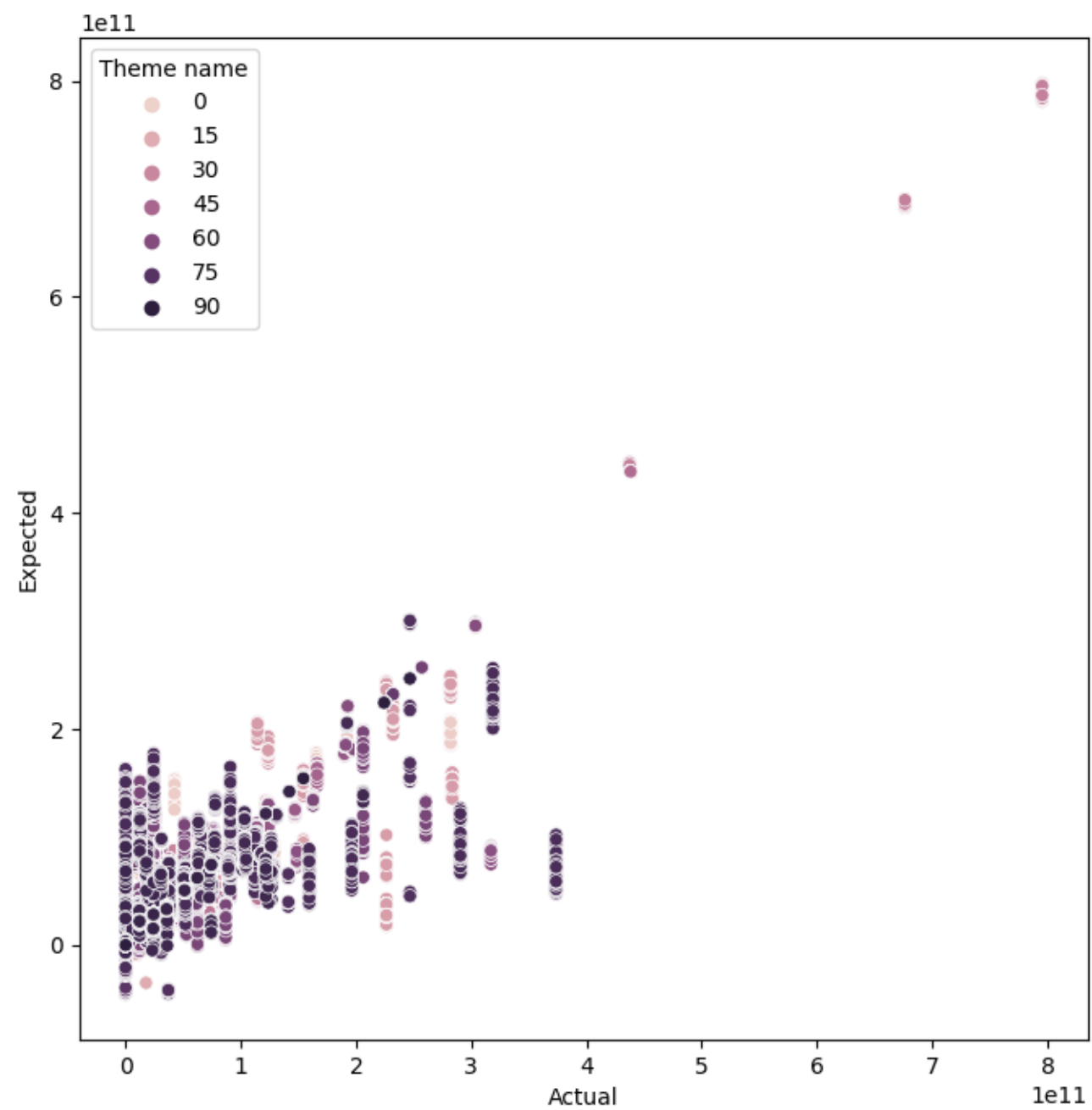


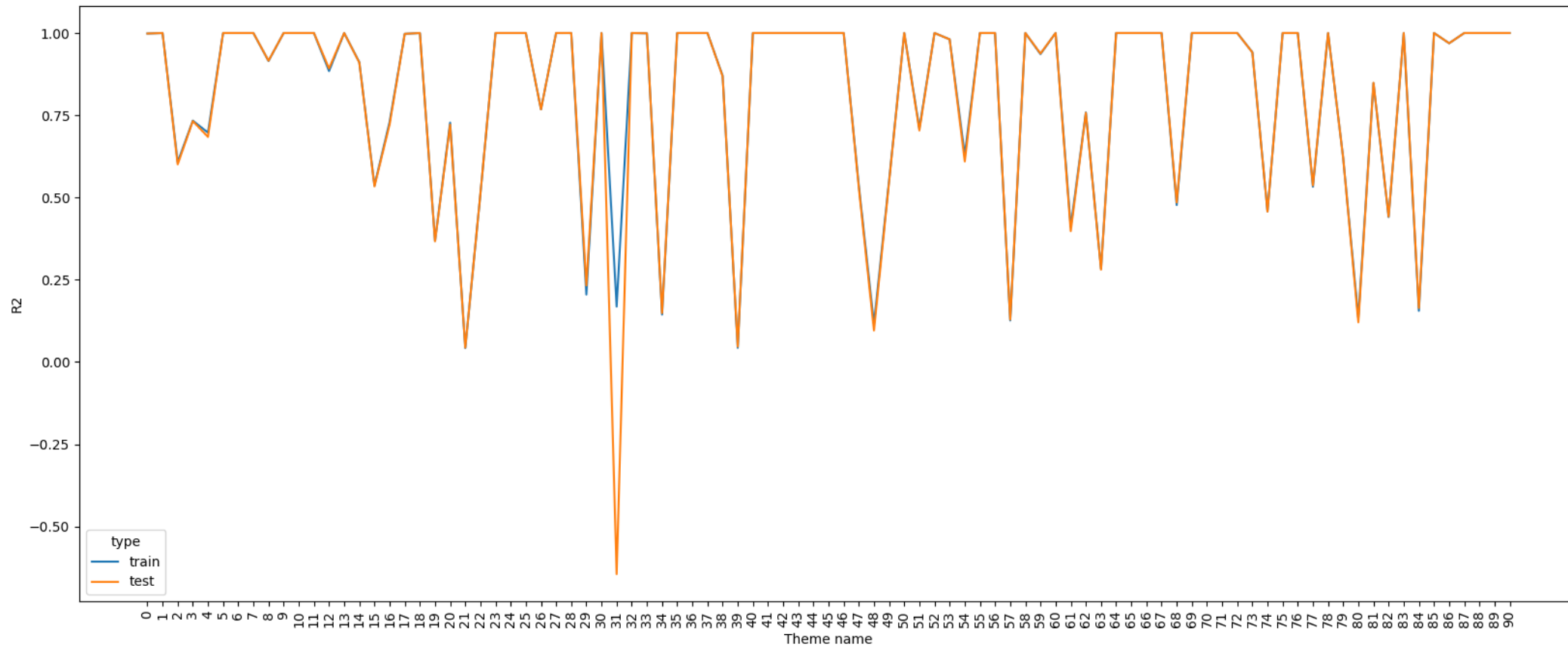
Модель Ridge

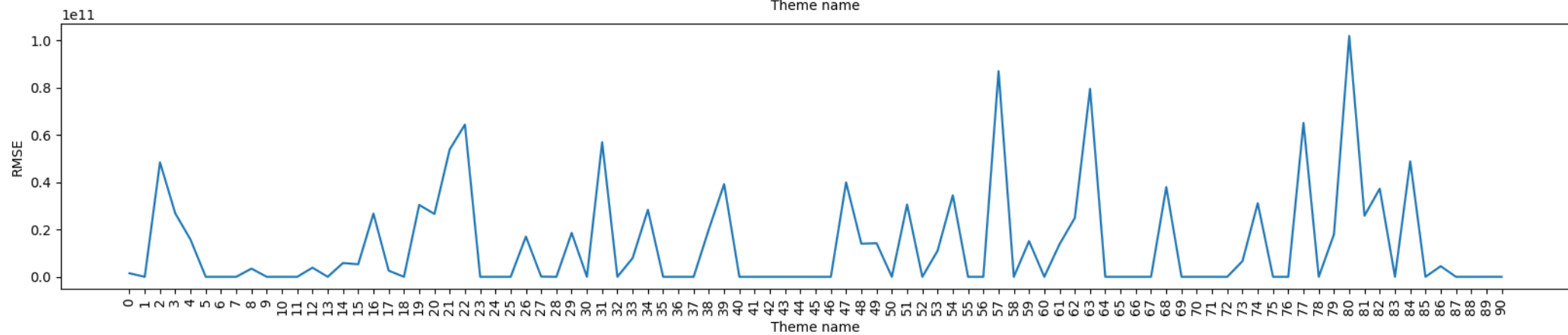
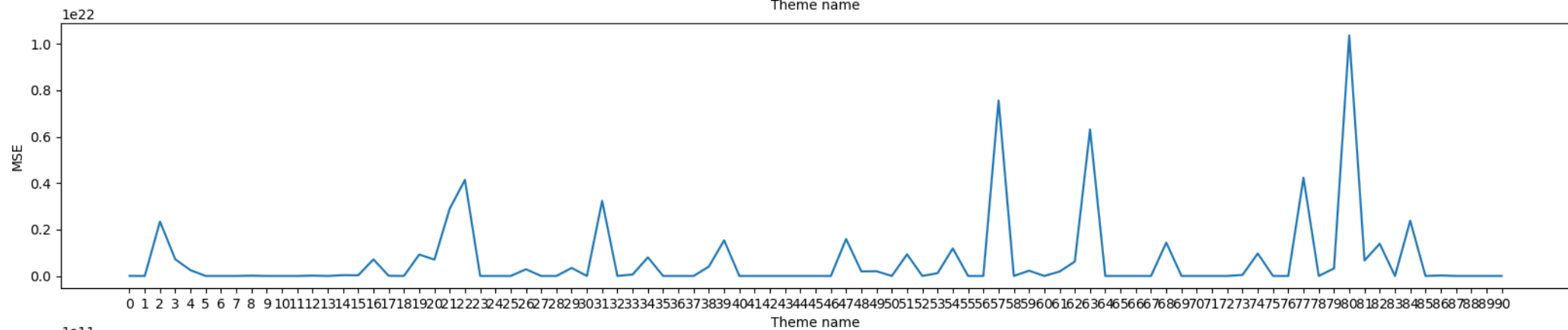
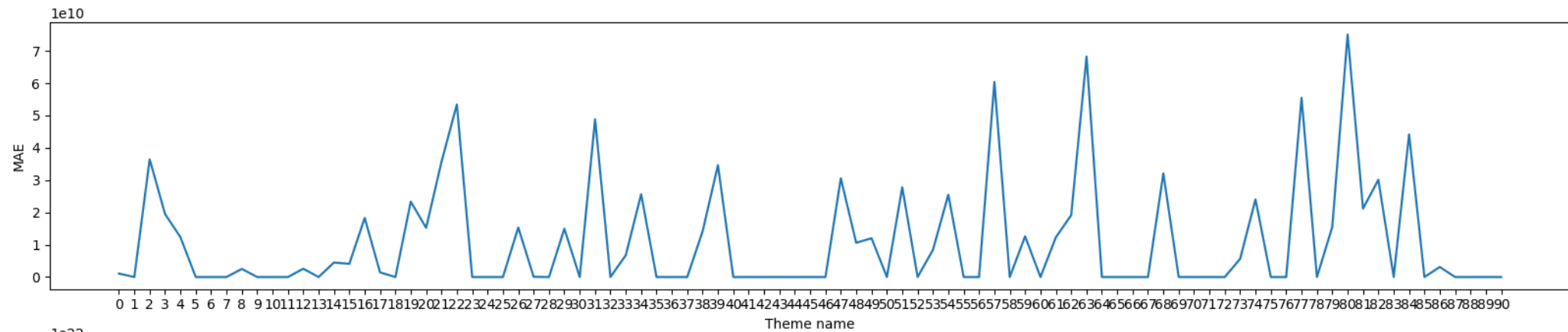
Ridge ⚡⚡⚡

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.
- **Однако** 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.





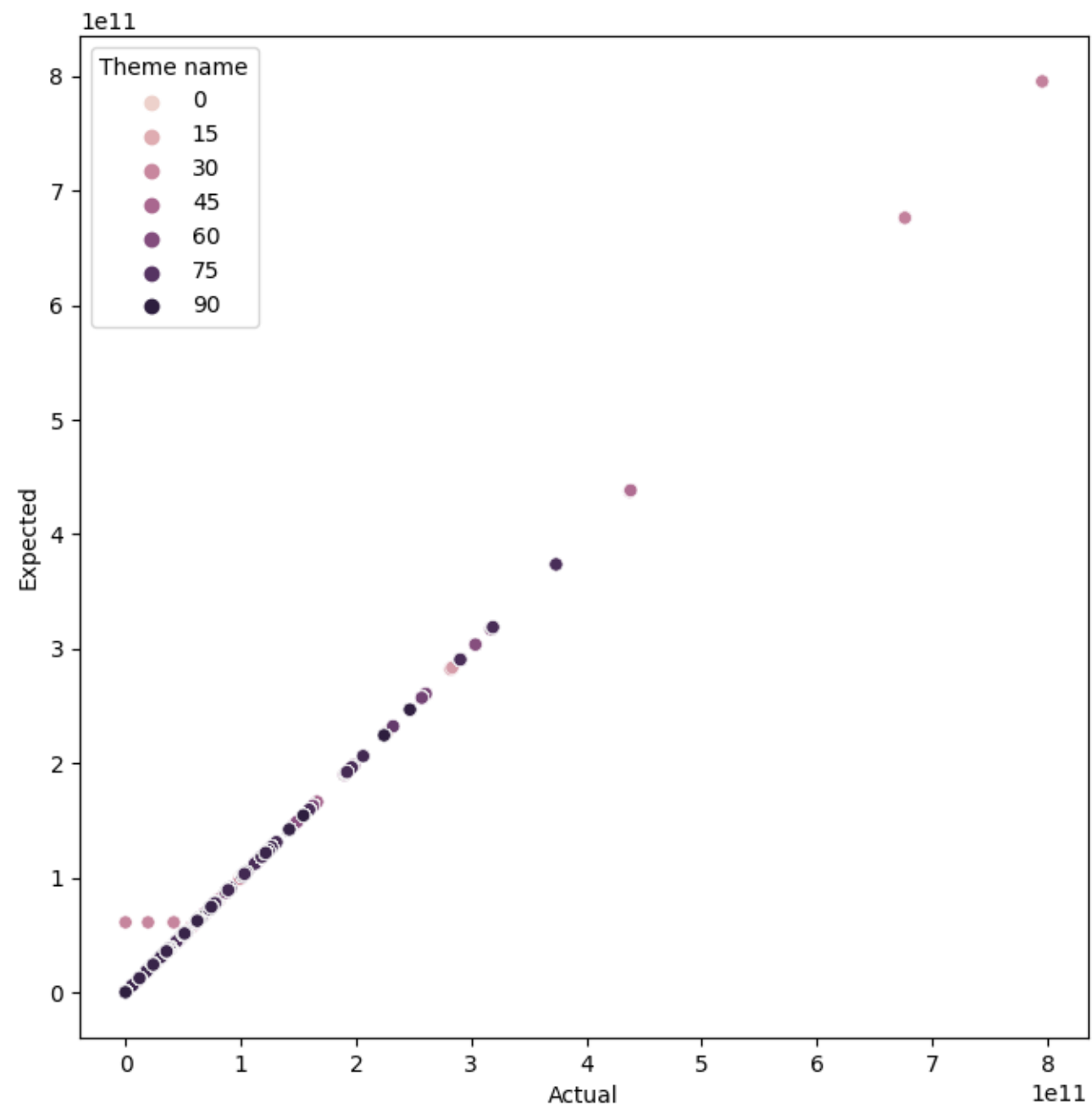


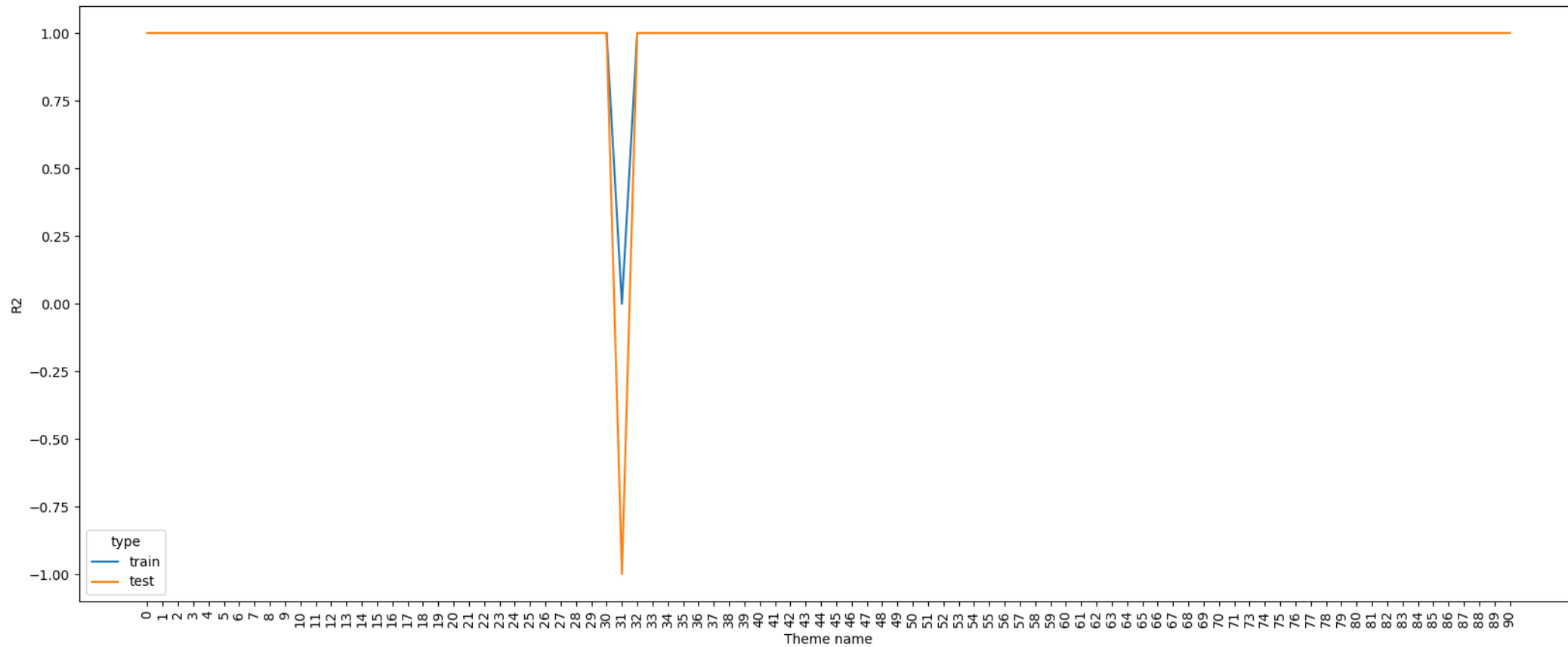
Модель RF

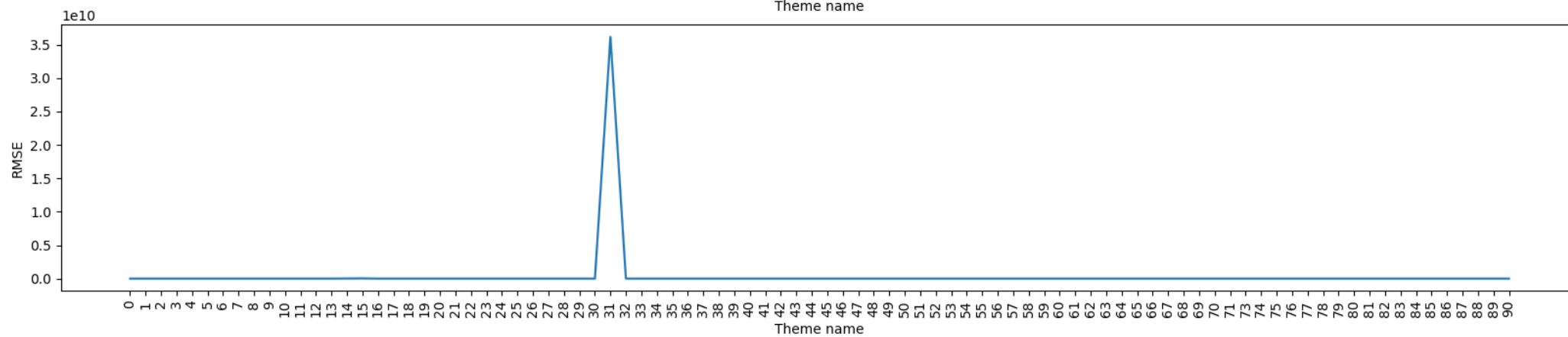
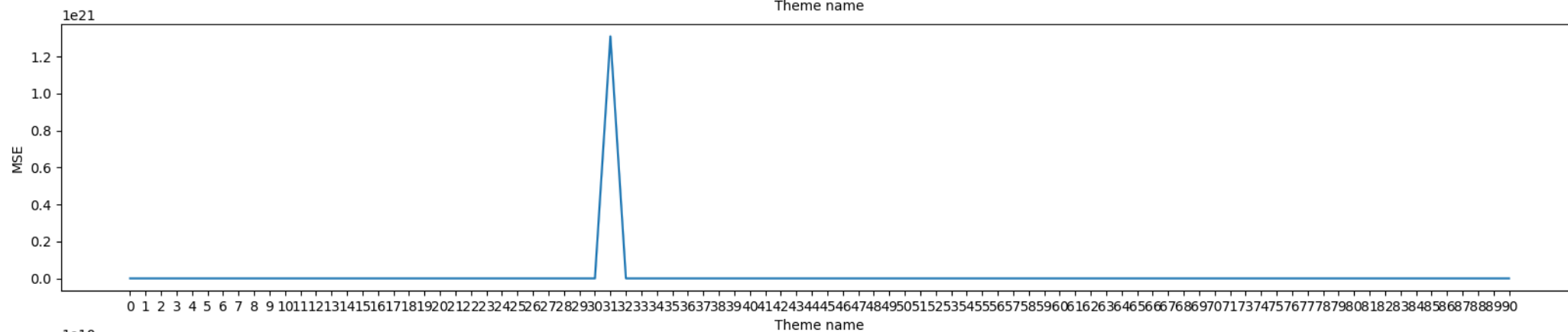
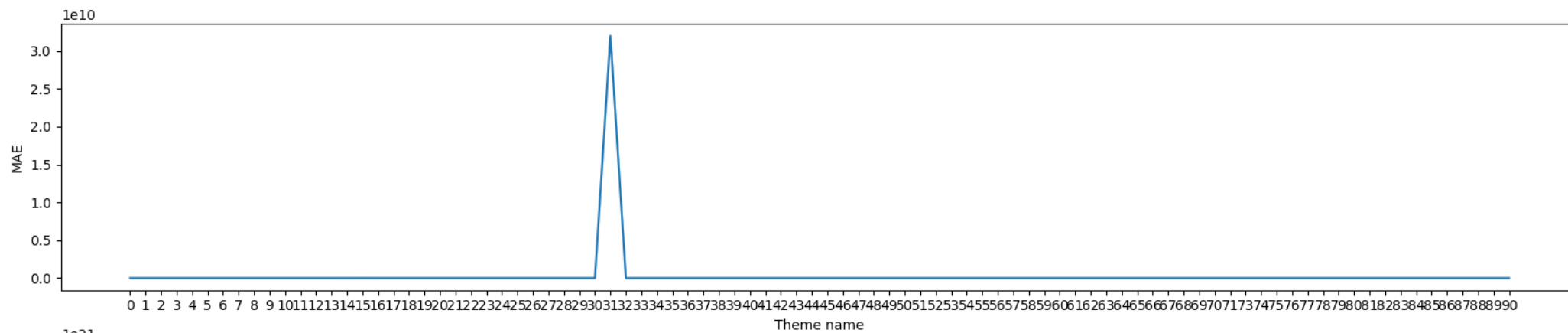
Random Forest 🌳🌳🌳

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.





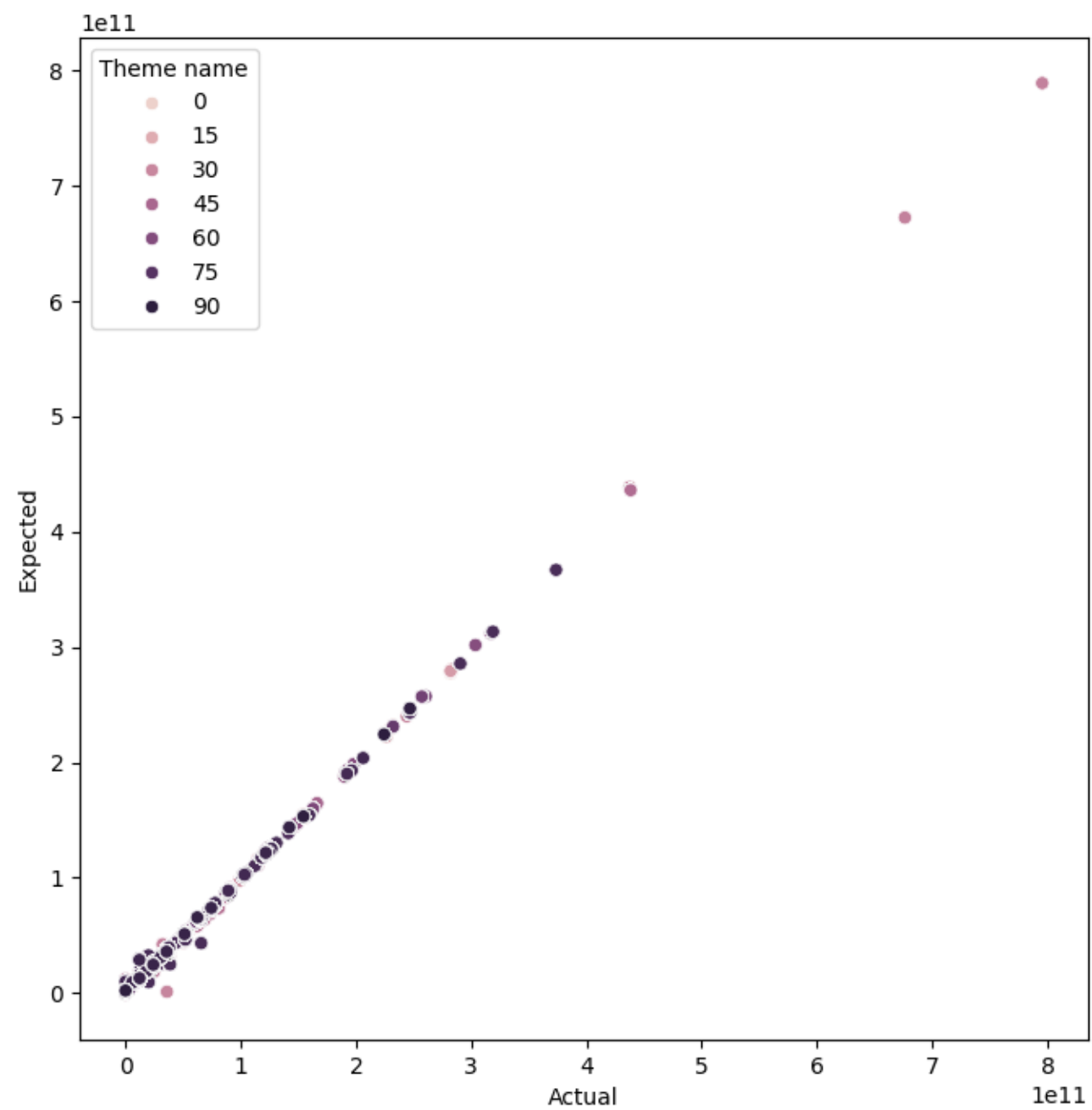


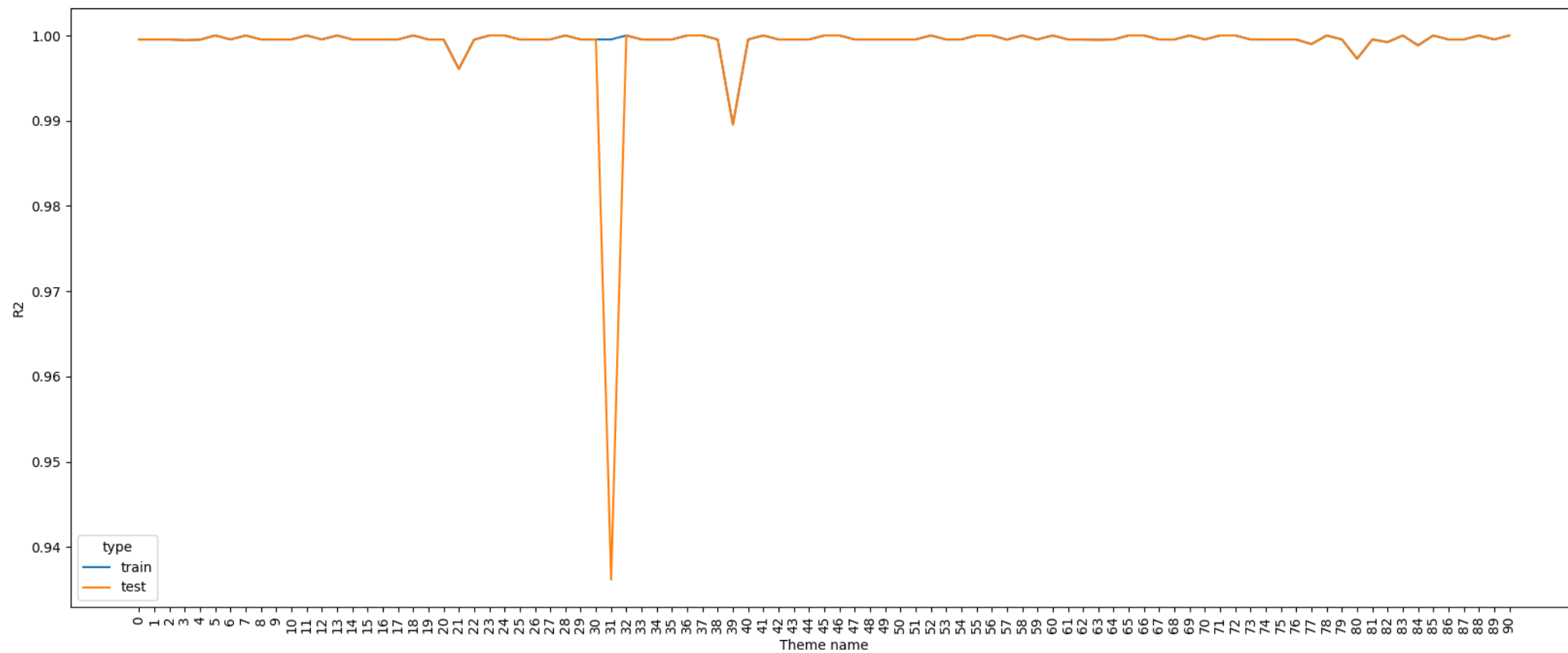
Модель GB

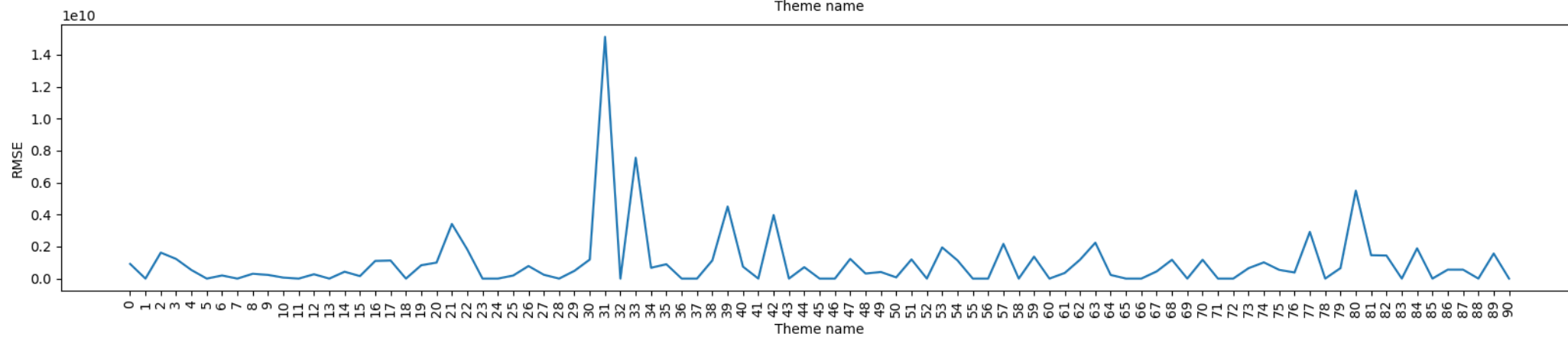
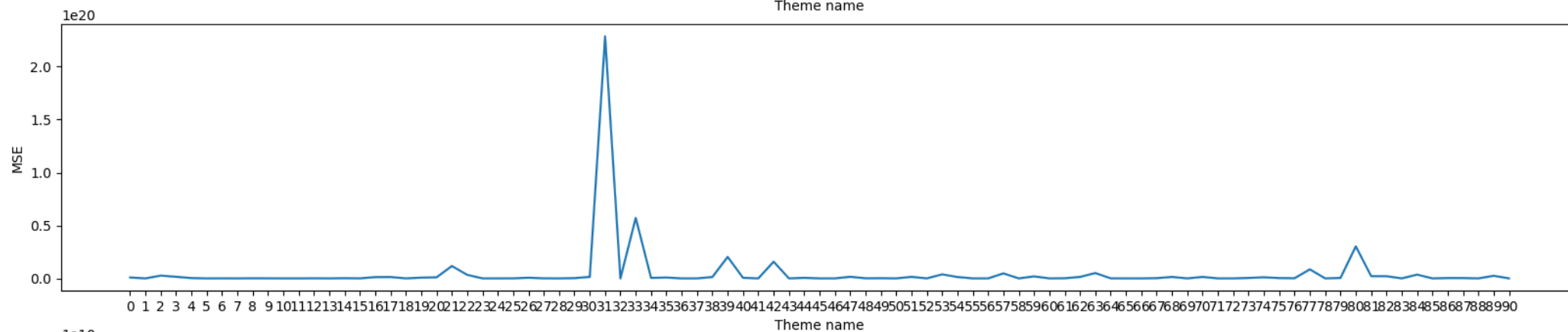
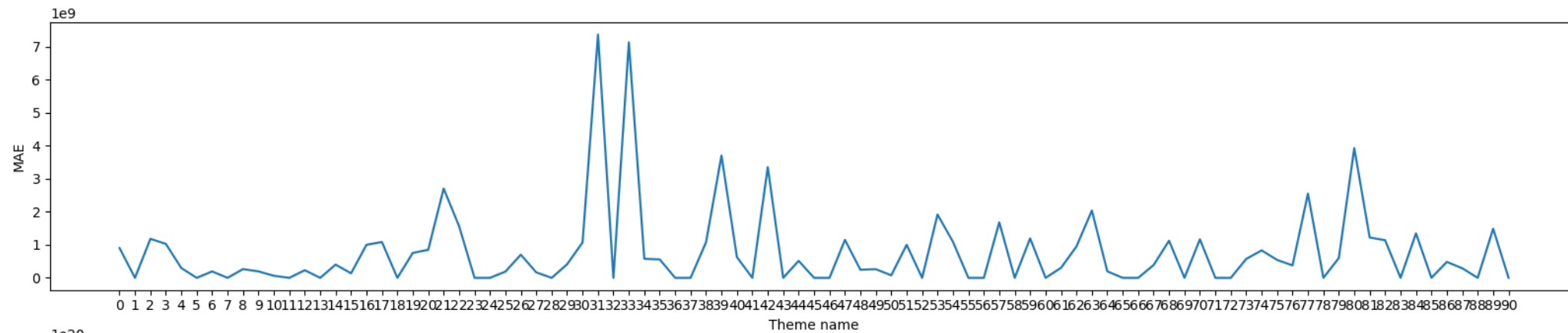
Gradient Boosting 

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.





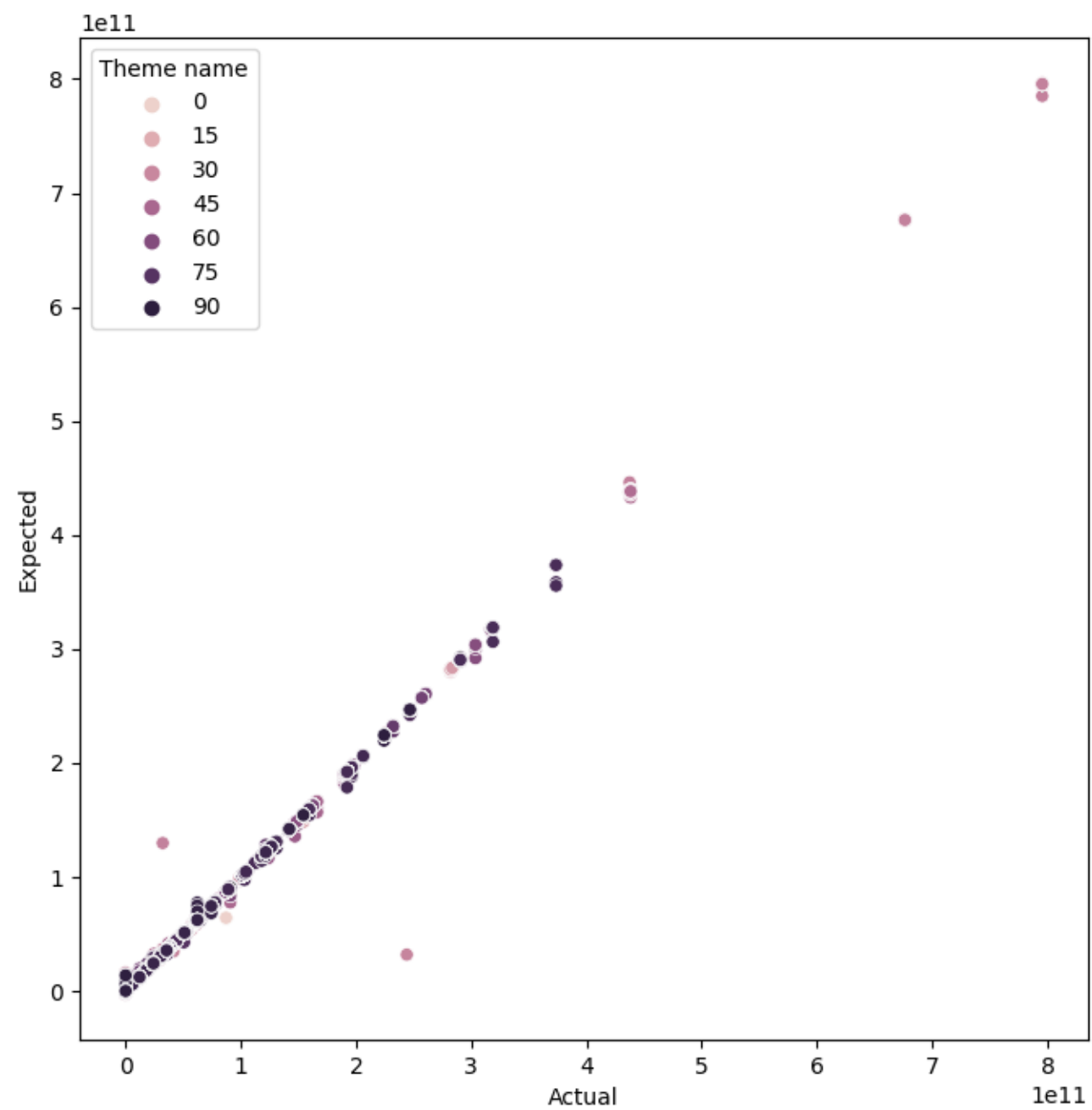


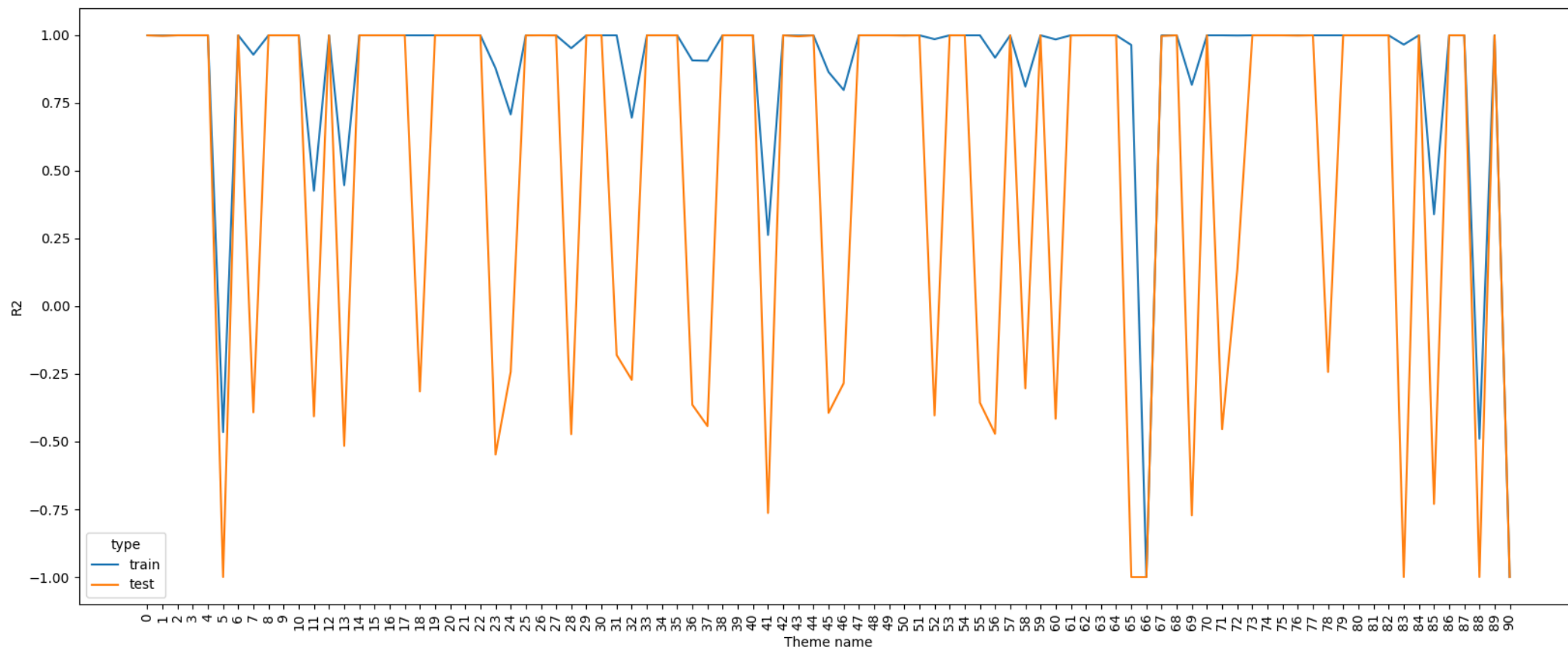
Модель СВ

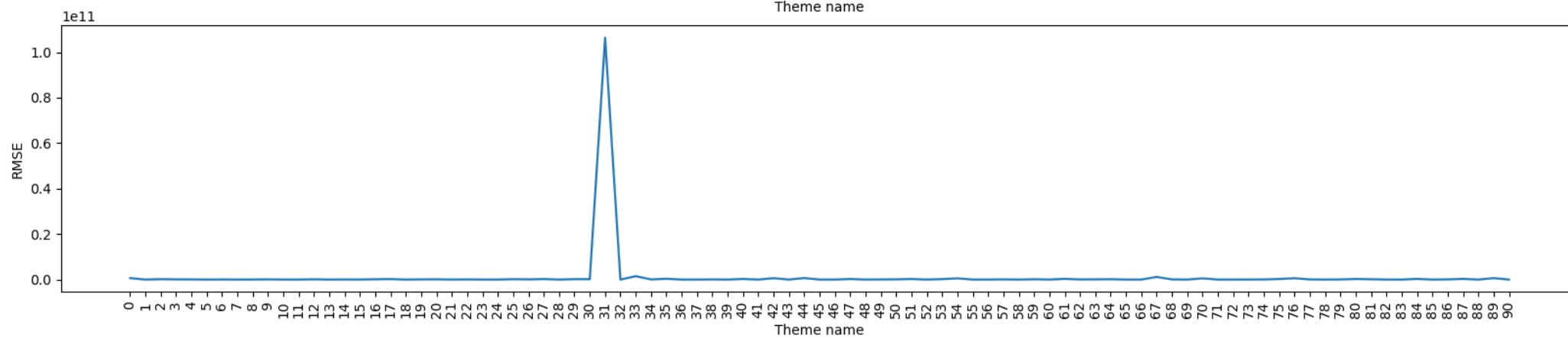
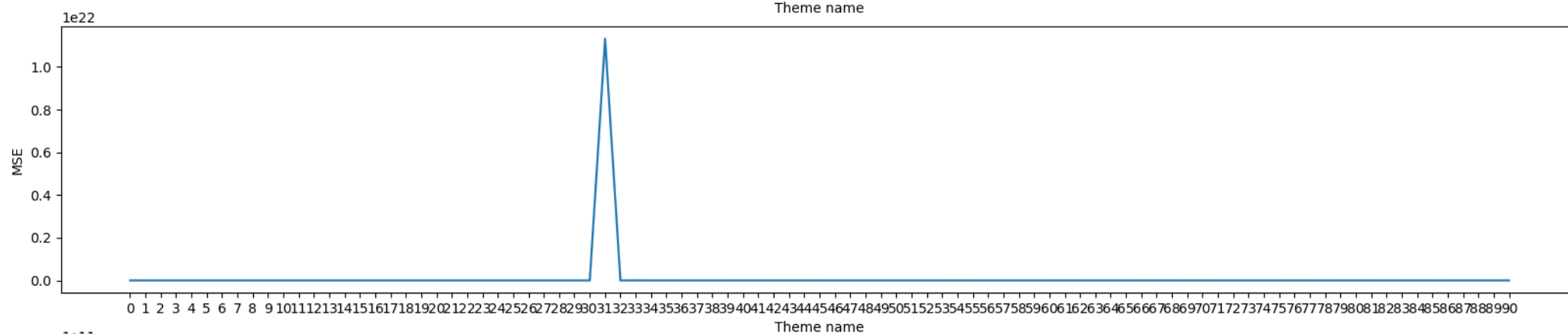
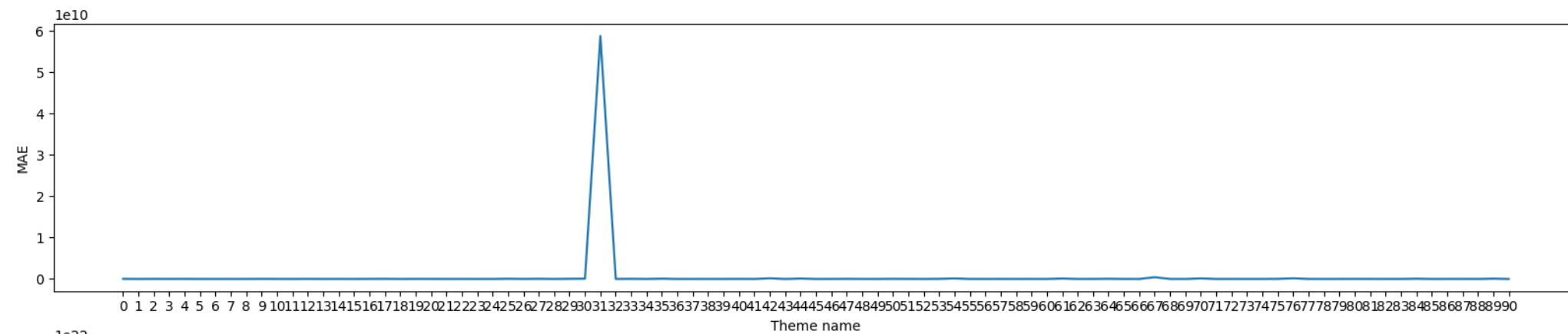
Cat Boost Regressor 🐱🐱🐱

СВ это регрессионная модель, входящая в состав библиотеки CatBoost, разработанной Yandex. CatBoost (Categorical Boosting) представляет собой алгоритм градиентного бустинга, оптимизированный для работы с категориальными данными.

- **Плюс:** он автоматически преобразует категориальные переменные и интегрирует их в модель, что делает его очень удобным и мощным для наборов данных с многочисленными категориальными признаками.
- **Но к сожалению,** CatBoost может требовать больше времени для обучения по сравнению с некоторыми другими алгоритмами машинного обучения, особенно на больших наборах данных.







Почему коту плохо?

Ответ убил. 💀💀💀

```
In 19 1 wtf = lego[lego['Theme name'] == 5][['Set Price']]
      2 wtf
```

Executed in 44ms, 23 Dec at 02:43:27

Out 19

10570 rows x 1 columns pd.DataFrame

CSV

	Set Price
77439	192264266667
77440	192264266667
77441	192264266667
77442	192264266667
77443	192264266667
77444	192264266667
77445	192264266667
77446	192264266667
77447	192264266667
77448	192264266667

```
In 20 1 len(wtf[wtf['Set Price'] == 192264266667])
```

Executed in 7ms, 23 Dec at 02:43:29

Out 20

10570

И так во многих...

Результаты 🏆🏆🏆

- **Random Forest** 🏆¹

Одно дерево хорошо, а лес лучше 🏕️

- **Gradient Boosting** 🏆²

Gradient Boosting это современный эффективный способ для моделирования регрессии 🌌

- **Cat Booster** 🏆³

Cat Boost Library это специализированный инструмент, разработанный в Yandex 💖 для решения комплексных задач 🐱

Работал,

Team Lead of Data Balbesing



Михаил Бабушкин КЭ-301

Я сладко сплю новогодние
праздники после закрытого
майнора ->

