Київський національний університет імені Тараса Шевченка

Фізичний факультет

Кафедра загальної фізики

**Звіт**

**про науково-дослідну практику із фізичного наноматеріалознавства**

Тема роботи : **«**Застосування символьної регресії до опису процесів у сонячних елементах**»**

Студента 1 курсу магістратури

освітньої програми «Фізика наносистем»

кафедри загальна фізика

Куща Івана Олександровича

Керівник практики:

Олег ОЛІХ, проф., д.фіз.-мат. наук

Науковий керівник:

Олег ОЛІХ, проф., д.фіз.-мат. наук

Київ – 2024

**ЗМІСТ**

графік виконання магстерської практики ………………….2

ВСТУП…………………………………………………………………………3

ОПИС МЕТОДУ PySRRegressor……………………………………………. 4

ЕКСПЕРИМЕНТ n1:оцінка моделі ……………………………………6

ЕКСПЕРИМЕНТ n2:оцінка моделі ……………………………………8

ЕКСПЕРИМЕНТ n3:оцінка моделі ……………………………………9

ЕКСПЕРИМЕНТ n4:оцінка моделі ……………………………………11

ВИСНОВОК……………………………………………………………………13

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ …………………………………16

План

науково-виробничої практики з фізики наноматеріалів Куща.І.О.

(ОКР “магістр” , І курс)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Вид роботи | Лаб. | Термін виконання |
| 1 | Знайомство з пакетом | 432 Кабінет  Лаб.Корпус Фіз.Фак. | 25.01.2024-17.02.2024 |
| 2 | Апроксимація залежності струму короткого замикання с.е. від довжини дифузії носіїв, | 432 Кабінет  Лаб.Корпус Фіз.Фак. | 17.02.2024-14.03.2024 |
| 3 | Аналіз температурної залежності ширини забороненої зони, | 432 Кабінет  Лаб.Корпус Фіз.Фак. | 14.03.2024-01.04.2024 |
| 4 | Опис температурної залежності струму насичення с.е. | 432 Кабінет  Лаб.Корпус Фіз.Фак. | 01.04.2024-20.04.2024 |
| 5 | Оптимізація моделей Символічної  Регресії | 432 Кабінет  Лаб.Корпус Фіз.Фак. | 20.04.2024-20.05.2024 |
| 6 | Підготовка звіту | Дист. | 20.05.2024-02.06.2024 |
|  |  |  |  |

Ознайомлений Кущ І.О.

Керівник науково-виробничої практики Оліх.О.Я

Вступ

Машинне навчання набуває все більшого значення у фізиці, допомагаючи вирішувати складні задачі, аналізувати великі обсяги даних та відкривати нові закономірності. Завдяки своїй здатності знаходити приховані патерни та оптимізувати процеси, методи машинного навчання вже активно застосовуються у різних галузях фізики, таких як квантова механіка, астрофізика та матеріалознавство. Наприклад, нейронні мережі використовуються для прогнозування поведінки складних систем, а методи кластеризації допомагають ідентифікувати нові фази матерії.

Однак, традиційні методи машинного навчання, такі як нейронні мережі або градієнтний бустинг, часто діють як "чорні ящики", надаючи мало інформації про фізичні взаємозв'язки між змінними. У фізиці, де інтерпретованість і розуміння процесів є ключовими, цього може бути недостатньо. Тут на допомогу приходить символічна регресія, яка дозволяє знаходити математичні вирази, що точно описують дані, зберігаючи при цьому прозорість і інтерпретованість.

Символічна регресія, зокрема через використання таких інструментів, як PySR, дає можливість вивести компактні, математично інтуїтивні моделі, які можуть бути легко перевірені на відповідність фізичним законам. Це робить її потужним інструментом для фізиків, які прагнуть не лише отримати точні прогнози, але й зрозуміти фундаментальні принципи, що лежать в основі досліджуваних явищ. Використовуючи символічну регресію, можна автоматично виявляти закони природи, зводячи складні дані до простих, зрозумілих формул, що значно сприяє науковим відкриттям і прогресу[2].

Моя магістерська практика протягом цього семестру була присвячена розробці та апроксимації математичних моделей за допомогою даних та пакету PySR. Основною метою було створення формул, які точно описують задані дані, використовуючи метод символічної регресії. Цей процес дозволяє не лише знаходити моделі, які добре апроксимують дані, але й зрозуміти підлягаючі закономірності та взаємозв'язки у даних.

PySR (Python Symbolic Regression) є потужним інструментом для виконання символічної регресії, яка полягає у пошуку математичних виразів, що найкраще описують вхідні дані. PySR використовує еволюційні алгоритми для генерування та оптимізації формул, що дозволяє знаходити компактні та інтерпретовані математичні моделі[3].

Основний компонент пакету - PySRRegressor, який надає інтерфейс для тренування та оцінки моделей символічної регресії. PySRRegressor автоматично шукає найкращу формулу шляхом ітеративного еволюційного процесу, що включає генерацію можливих виразів, їх оцінку та вибір найкращих серед них на основі заданих критеріїв, таких як точність та складність моделі.

Протягом семестру я працював над кількома наборами даних, застосовуючи PySRRegressor для розробки точних та інтерпретованих моделей. Цей процес включав попередню обробку даних, налаштування параметрів PySRRegressor, а також аналіз отриманих результатів для забезпечення найкращої можливої апроксимації даних. Завдяки цьому я здобув глибокі знання в області символічної регресії та її практичного застосування для аналізу даних.

Опис методу PySRRegressor

PySRRegressor є ключовим компонентом пакету PySR, призначеним для виконання символічної регресії, що дозволяє знаходити математичні вирази, які точно описують дані. Символічна регресія відрізняється від традиційних методів регресії тим, що не обмежується попередньо визначеними функціональними формами (наприклад, лінійними або поліноміальними моделями). Натомість вона автоматично вивчає структуру даних і генерує математичні вирази, які можуть включати будь-які комбінації математичних операторів[1].

Основні характеристики та можливості PySRRegressor:

1. Еволюційні алгоритми: PySRRegressor використовує еволюційні алгоритми для пошуку оптимальних математичних виразів. Еволюційні алгоритми імітують процес природного відбору, генеруючи численні потенційні рішення (математичні формули) та еволюціонуючи їх через операції схрещування, мутації та селекції[2].

2. Генерація формул: PySRRegressor генерує формули, комбінуючи базові математичні оператори, такі як додавання, віднімання, множення, ділення, а також більш складні функції, як-от експоненти, логарифми, тригонометричні функції тощо. Це дозволяє створювати широкий спектр можливих моделей.

3. Оцінка моделей: Кожна згенерована формула оцінюється на основі точності (як добре вона підходить до навчальних даних) та складності (наскільки проста або складна формула). PySRRegressor балансує між точністю та інтерпретованістю, прагнучи знайти моделі, які не тільки добре апроксимують дані, але й легко інтерпретуються[2].

4. Автоматичне налаштування параметрів: PySRRegressor має можливість автоматично налаштовувати свої параметри, що включає розмір популяції, кількість поколінь, ймовірності мутацій і схрещування, та інші параметри еволюційного алгоритму. Це робить процес пошуку моделей більш ефективним та автоматизованим.

5. Інтеграція з іншими бібліотеками: PySRRegressor легко інтегрується з іншими популярними бібліотеками для аналізу даних та машинного навчання, такими як NumPy, Pandas, та Scikit-learn. Це дозволяє використовувати PySR у складі більш складних аналітичних робочих процесів.

6. Паралельні обчислення: PySR підтримує паралельні обчислення, що дозволяє значно прискорити процес пошуку оптимальних моделей на великих наборах даних.

|  |
| --- |
| Symbolic regression - Wikipedia |
| Рис. 1. Схематичне зображення роботи символічної регресії  За допомогою еволюційних алгоритмів |

# Процес роботи з PySRRegressor:

1. Підготовка даних: Вхідні дані потрібно підготувати та розділити на навчальну та тестову вибірки[3].

2. Ініціалізація PySRRegressor: Створення об'єкта PySRRegressor з налаштуванням початкових параметрів, таких як список базових функцій, максимальна кількість термів у формулі та інші параметри.

3. Тренування моделі: Використання методу `fit` для тренування PySRRegressor на навчальних даних. Еволюційний алгоритм генерує та оптимізує математичні вирази.

4. Оцінка результатів: Після завершення тренування оцінюються згенеровані моделі на тестових даних. Вибирається найкраща модель на основі компромісу між точністю та складністю.

5. Інтерпретація та використання моделі: Обрана модель аналізується та використовується для прогнозування або інтерпретації даних.

# Процес Вибору Рівняння в Символічній Регресії

Вибір рівняння виконується вручну наступним чином. Спочатку результат обчислень складається з кількох рівнянь-кандидатів різного рівня складності. Складність (КОМП) оцінюється шляхом підрахунку кількості входжень кожного оператора, константи та вхідної змінної. Спочатку ми вибираємо кілька рівнянь, для яких спостерігається часткове зменшення середньої квадратичної помилки, обраної як функція втрат, порівняно зі збільшенням складності наступної найкращої моделі. Будь-яке рівняння, що має відносну похибку щодо даних моделювання меншу за заздалегідь визначене порогове значення (наприклад, 1%), розглядається як потенційна аналітична модель, і нарешті вибирається модель із найменшою складністю. Експериментально встановлено, що таким чином виробляються найкращі рішення, які відповідають цілям досліджуваної проблеми.[3]

Символічна регресія (SR) може призвести до складних виразів, які важко інтерпретувати і які містять небажані функції, такі як вкладені операції. З цієї причини застосовуються обмеження, щоб контролювати вкладеність операторів, не дозволяючи вкладеність тих самих операторів взагалі, якщо не вказано інше. Для не великих наборів даних і в тих випадках, коли всі точки даних не потрібні, зазвичай використовується вибірка з приблизно 300 точок. Виявлено, що випадкових вибірок від 100 до 600 точок даних більш ніж достатньо для отримання математичних рівнянь на основі SR. Решта даних використовується для порівняння з кінетичними результатами як набір тестування, щоб забезпечити точність рівнянь SR для всього набору даних. Навчання моделі SR повторюється кілька разів (зазвичай близько 10 разів) з використанням різних випадкових вибірок. Завжди отримується приблизно одне й те саме каркасне рівняння, що додатково забезпечує валідність поточної моделі SR.

Для цілей цієї роботи ми очікуємо отримати точні аналітичні рівняння на основі SR, які можна легко використовувати в інженерних розрахунках. Досягається компроміс між точністю, складністю та загальною кількістю запропонованих рівнянь, одночасно забезпечуючи валідність рівнянь для відповідного діапазону параметрів.[2]

Експеримент N1: Оцінка Моделі PySRRegressor

В рамках експерименту N1 було використано модель PySRRegressor з наступними параметрами: метод вибору моделі - "best", кількість популяцій - 120, максимальний розмір виразу - 120, розмір популяції - 120, максимальна глибина дерева - 7. Параметр parsimony встановлено на рівні 0.0001, що допомагає контролювати складність моделі, а параметр weight\_optimize - на 0.001, що впливає на оптимізацію ваг формули. Використовувались основні бінарні оператори: додавання, віднімання, ділення та множення, з обмеженням на вкладеність оператора ділення. Модель була навчена на двох процесорах.

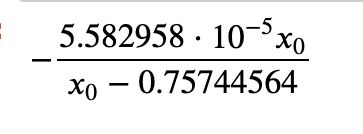


Рис.2 Вибрана апроксимуюча формула

На основі тренувального набору

# Результати Оцінки Моделі

Модель PySRRegressor була навчена на наборі даних, і результати оцінки показали високу точність моделі. Середня абсолютна помилка (MAE) склала 1.507404383048917 \* 10^(-8), що вказує на дуже малу середню різницю між передбаченими та фактичними значеннями. Середньоквадратична помилка (MSE) склала 0.00011092515522408424, що також свідчить про високу точність моделі та малу кількість помилок у передбаченнях.

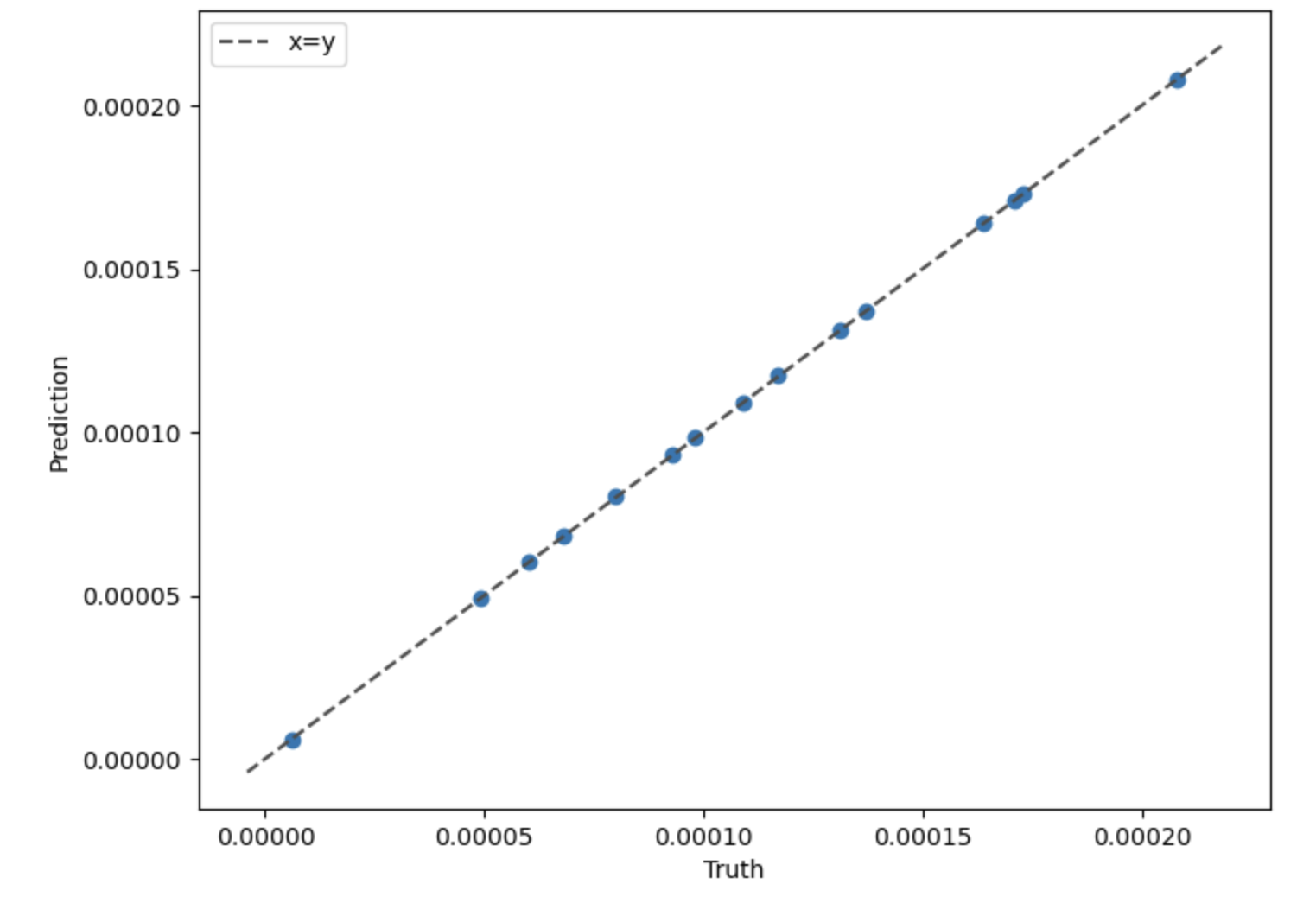


Рис.3 Графік порівняння передбачених

Та фактичних значень експерименту N1

Ці результати підтверджують ефективність налаштованої моделі PySRRegressor у знаходженні точних математичних виразів для апроксимації даних. Високі показники точності та контрольована складність моделі свідчать про те, що обрані параметри добре збалансовані, що дозволяє отримувати інтерпретовані та точні аналітичні моделі.

Експеримент N2: Оцінка Моделі PySRRegressor

В рамках експерименту N2 було використано модель PySRRegressor з наступними параметрами: метод вибору моделі - "best", кількість популяцій - 50, максимальний розмір виразу - 50, розмір популяції - 50, максимальна глибина дерева - 5. Параметр parsimony встановлено на рівні 0.001, що допомагає контролювати складність моделі, а параметр weight\_optimize - на 0.001, що впливає на оптимізацію ваг формули. Використовувались основні бінарні оператори: додавання, віднімання, ділення та множення, а також унарний оператор `square`. Модель була навчена на п'яти процесорах.



Рис.4 Вибрана формула для апроксимації

Другого набору данних

# Результати Оцінки Моделі

Модель PySRRegressor була навчена на наборі даних, і результати оцінки показали високу точність моделі. Середня абсолютна помилка (MAE) склала 8.15103567617 \*10^(-9), що вказує на дуже малу середню різницю між передбаченими та фактичними значеннями. Середньоквадратична помилка (MSE) склала 0.00123433765982598, що також свідчить про високу точність моделі та малу кількість помилок у передбаченнях.

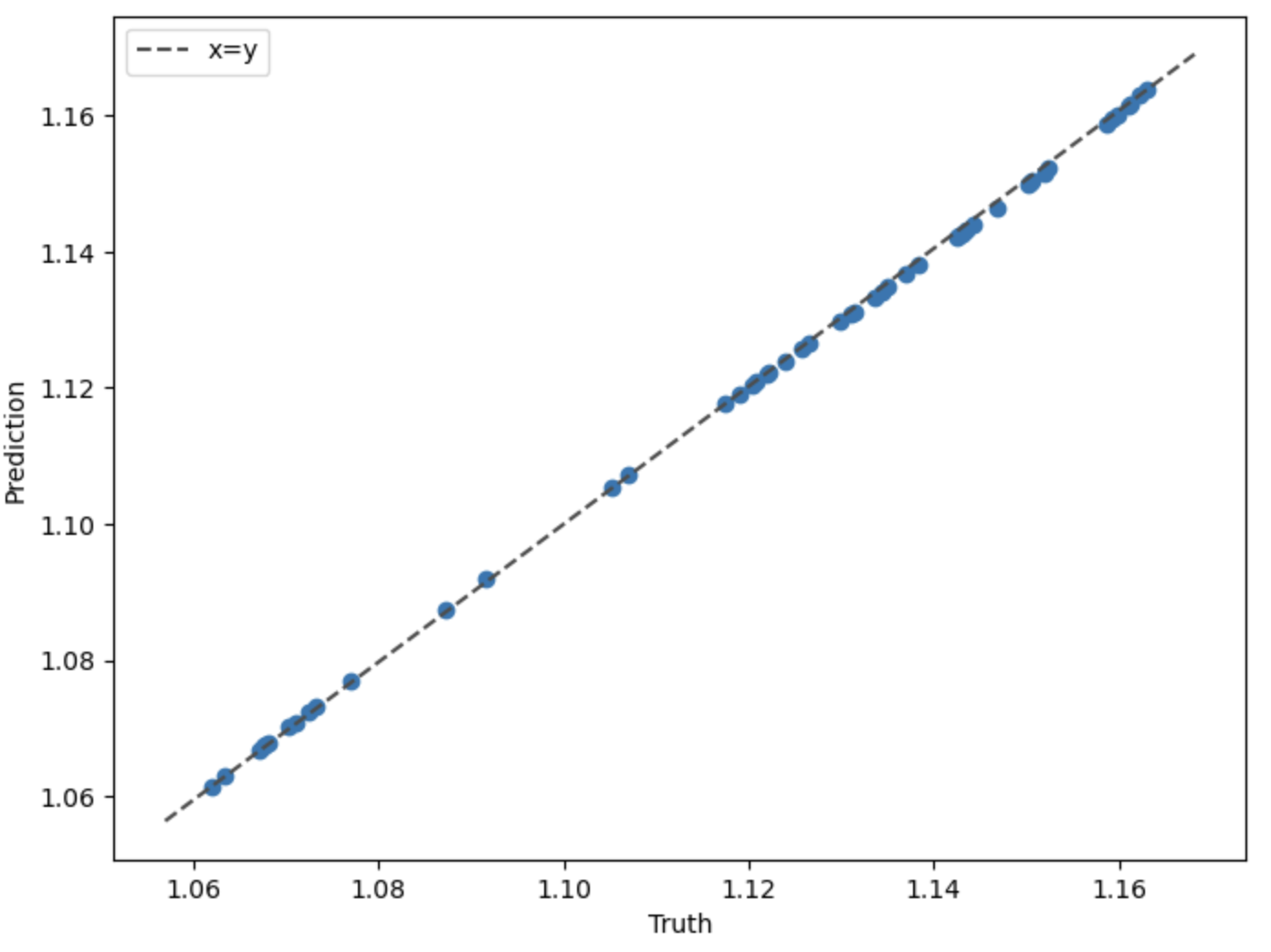


Рис.5 Графік порівняння передбачених

Та фактичних значень експерименту N2

Ці результати підтверджують ефективність налаштованої моделі PySRRegressor у знаходженні точних математичних виразів для апроксимації даних. Високі показники точності та контрольована складність моделі свідчать про те, що обрані параметри добре збалансовані, що дозволяє отримувати інтерпретовані та точні аналітичні моделі.

Експеримент N3: Оцінка Моделі PySRRegressor з Власним Бінарним Оператором

В рамках експерименту N3 було використано модель PySRRegressor з наступними параметрами: метод вибору моделі - "best", максимальна глибина дерева - 100, максимальний розмір виразу - 100, кількість популяцій - 100, розмір популяції - 100, а також використання 12 процесорів для паралельних обчислень. Особливою характеристикою цього експерименту є створення та використання власного бінарного оператора `ExpDiv`, який визначається як `exp(y/x)`. Також були використані стандартні бінарні оператори: додавання, віднімання та множення, а також унарний оператор `cube`.

# Створення Власного Бінарного Оператора `ExpDiv`

Бінарний оператор `ExpDiv` визначається за допомогою наступного синтаксису: `'ExpDiv(x,y) = exp(y/x)'`. Додатково було вказано відображення для SymPy через параметр `extra\_sympy\_mappings`, який визначає функцію `ExpDiv` як `lambda x,y: exp(y/x)`. Для обмеження складності операцій були задані наступні параметри: `ExpDiv` може бути використаний до 9 разів як бінарний оператор і до 2 разів як унарний, а оператор `cube` - лише один раз. Вкладеність операторів була також обмежена: `ExpDiv` і `cube` не можуть бути вкладені один в одного або самі в себе.

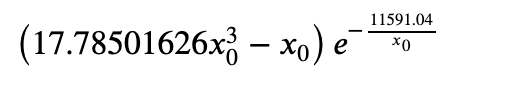


Рис.6 Вибрана модель що апроксимує данні експерименту N3

Із використанням вложенної функції

# Результати Оцінки Моделі

Модель PySRRegressor була навчена на навчальних даних, і результати оцінки показали високу точність моделі. Середньоквадратична помилка (MSE) склала 1.793718813654763 \*10^(-11), що свідчить про високу точність моделі та дуже малу кількість помилок у передбаченнях.

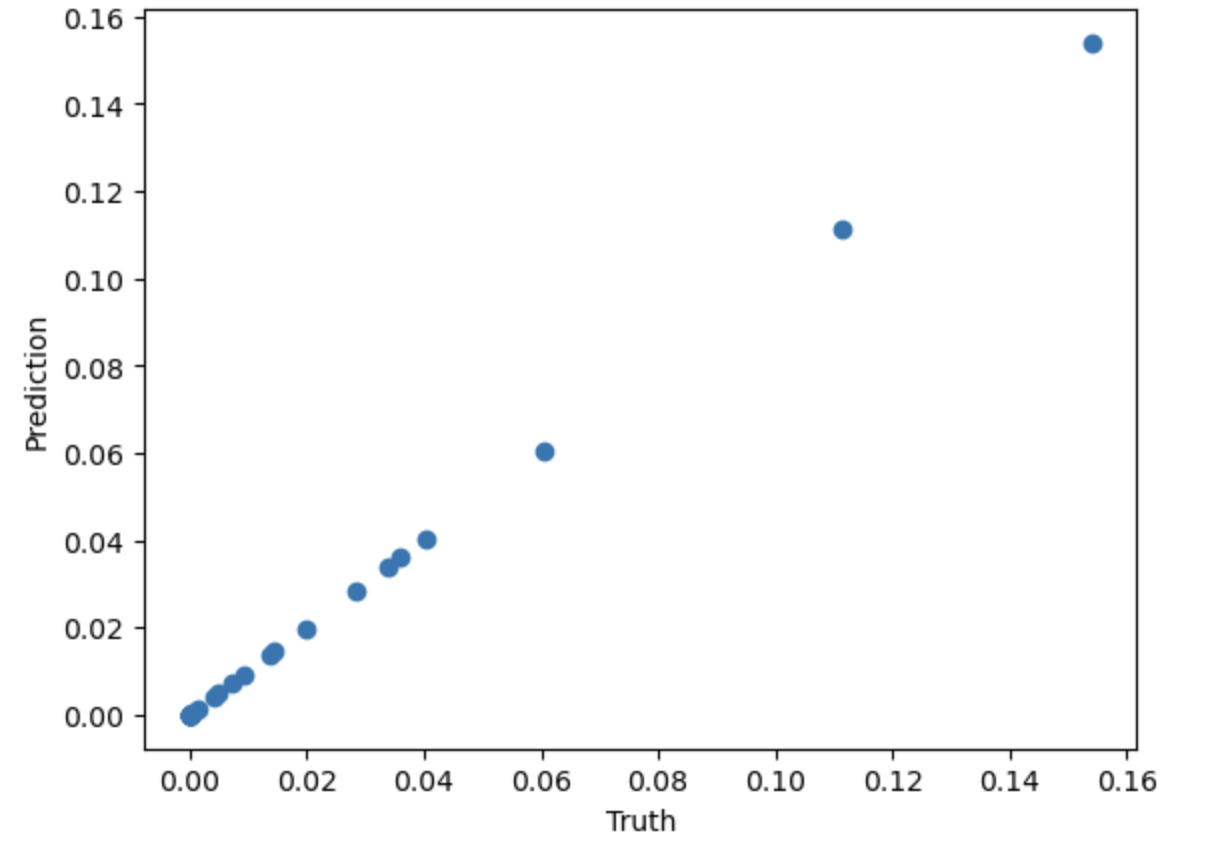


Рис.7 Графік порівняння передбачених

Та фактичних значень експерименту N3

Ці результати підтверджують ефективність налаштованої моделі PySRRegressor у знаходженні точних математичних виразів для апроксимації даних. Використання власного бінарного оператора `ExpDiv` показало, що модель здатна інтегрувати складніші математичні функції, зберігаючи високу точність та інтерпретованість результатів. Обрані параметри забезпечили баланс між точністю та складністю моделі, що дозволяє отримувати інтерпретовані та точні аналітичні моделі, придатні для подальшого використання в наукових та інженерних розрахунках.

Експеримент N4: Оцінка Моделі PySRRegressor із Залежністю Від Двох Змінних

В рамках експерименту N4 було використано модель PySRRegressor з наступними параметрами: метод вибору моделі - "best", кількість популяцій - 120, максимальний розмір виразу - 120, розмір популяції - 120, максимальна глибина дерева - 7. Параметр parsimony встановлено на рівні 0.0001, що допомагає контролювати складність моделі, а параметр weight\_optimize - на 0.001, що впливає на оптимізацію ваг формули. Використовувались основні бінарні оператори: додавання, ділення та множення, а також унарний оператор `neg`. Модель була навчена на двох процесорах.

# Особливість Експерименту: Залежність Від Двох Змінних

Основною відмінністю цього експерименту від попередніх трьох є те, що вперше залежна змінна визначається як функція двох інших змінних, а не однієї. Це дозволяє моделі враховувати більш складні взаємодії між змінними та знаходити більш точні математичні вирази для апроксимації даних.

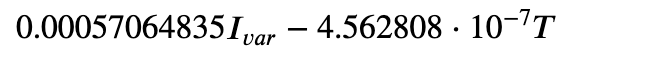


Рис.8 Вибрана модель що апроксимує данні експерименту N4

Зі декількома незалежними змінними

# Результати Оцінки Моделі

Модель PySRRegressor була навчена на наборі даних, і результати оцінки показали високу точність моделі. Середньоквадратична помилка (MSE) склала 1.396312 \*10^(-9), що свідчить про високу точність моделі та дуже малу кількість помилок у передбаченнях.

Ці результати підтверджують ефективність налаштованої моделі PySRRegressor у знаходженні точних математичних виразів для апроксимації даних, особливо коли залежність включає більше однієї змінної. Обрані параметри забезпечили баланс між точністю та складністю моделі, що дозволяє отримувати інтерпретовані та точні аналітичні моделі, придатні для подальшого використання в наукових та інженерних розрахунках. Використання моделі з двома незалежними змінними відкриває нові можливості для аналізу більш складних систем та взаємодій.

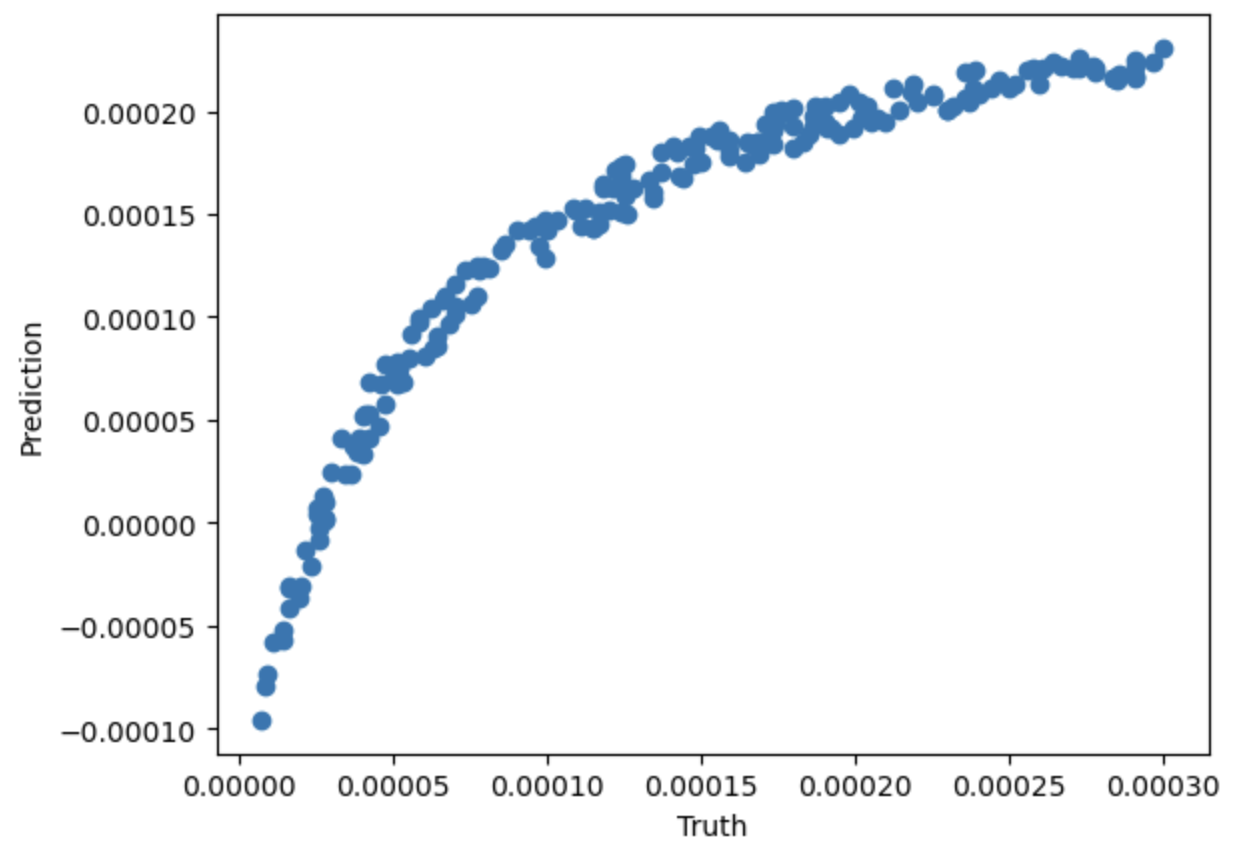


Рис.9 Графік порівняння передбачених

Та фактичних значень експерименту N4.

Висновок

Протягом цього дослідницького семестру було проведено чотири експерименти з використанням пакету PySR для символічної регресії з метою знаходження точних математичних виразів для апроксимації даних. Кожен експеримент був спрямований на оптимізацію параметрів моделі та оцінку її точності, що дозволило отримати цінні результати та висновки.

В першому експерименті модель PySRRegressor показала високу точність з середньою абсолютною помилкою (MAE) (1.507404383048917 \*10^{-8}) та середньоквадратичною помилкою (MSE) 0.00011092515522408424. Це продемонструвало ефективність обраних параметрів та здатність моделі до точного передбачення.

Другий експеримент також виявився успішним, з MAE (8.15103567617 \*10^(-9)) та MSE 0.00123433765982598, підтверджуючи здатність моделі до точної апроксимації даних при використанні унарного оператора `square` та основних бінарних операторів.

Третій експеримент був особливим завдяки використанню власного бінарного оператора `ExpDiv`, що показало гнучкість PySR у створенні спеціалізованих математичних виразів. Результат з MSE (1.793718813654763 \*10^(-11)) підкреслив високу точність моделі та її здатність до інтеграції складніших функцій.

Четвертий експеримент був першим, де залежна змінна визначалась від двох інших змінних, що дозволило моделі враховувати складніші взаємодії між змінними. MSE (1.396312 \* 10^(-9)) свідчить про високу точність та адаптивність моделі в аналізі більш складних систем.

Загалом, використання PySR для символічної регресії показало високу ефективність у знаходженні точних математичних виразів для різних наборів даних. Оптимізація параметрів моделі, використання спеціалізованих операторів та врахування складних взаємодій між змінними дозволили досягти високої точності та інтерпретованості результатів. Ці дослідження підтверджують потенціал PySR як потужного інструменту для наукових та інженерних розрахунків, де важлива точність та зрозумілість математичних моделей.

список використаної літератури

[1]. Sofos F., Dritselis C., Misdanitis S., Karakasidis T., Valougeorgis D. Computation of flow rates in rarefied gas flow through circular tubes via machine learning techniques. 2023.

[2]. Kronberger G., Burlacu B., Kommenda M., Winkler S. M., Affenzeller M. Symbolic Regression. Chapman & Hall, 2024. 312 с.

[3]. PySR: High-Performance Symbolic Regression in Python and Julia. Available at: <https://astroautomata.com/PySR/> (accessed June 3, 2024).