

# EcoMeow

NUNTANUT POONPAYAP, THAD CHOYRUM  
KRITTANIK SRITHANASARN

16TH NAPROCK PROGRAMMING CONTEST

# エコーミャウ



SOFTWARE FOR PREDICTION CO<sub>2</sub> EMISSIONS & ENERGY CONSUMPTION WITH MACHINE LEARNING  
ON REAL TIME COMPUTING AND ENGAGE AND EDUCATE THE PUBLIC ON GLOBAL WARMING



## Introduction

## 紹介

The rising levels of CO<sub>2</sub> due to human activities are accelerating climate change, leading to more frequent natural disasters and environmental degradation. In response, we propose "EcoMeow", a software system designed to predict CO<sub>2</sub> emissions and energy consumption based on the assumption of a growing global population, using machine learning techniques and real-time prediction. Additionally, the system features a cat mascot to engage and educate users on global warming and sustainability. Combining prediction models, real-time data, and educational content with an interactive mascot into one platform makes EcoMeow different from others.

## Problems

## 問題

The Goals set by the IPCC (Intergovernmental Panel on Climate Change) are as follows:

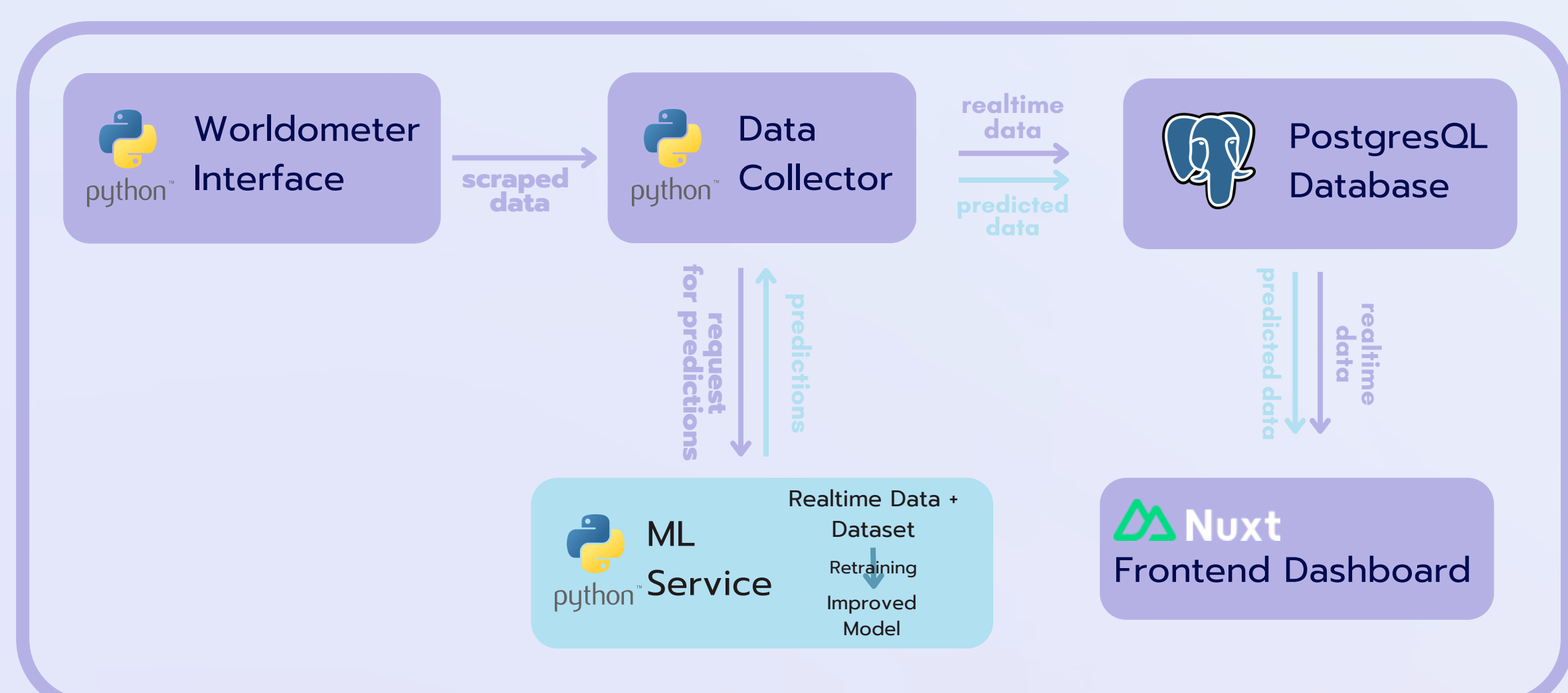
- By 2030:
  - Limiting Global Warming to 1.5°C Worldwide
  - 45% Reduction in CO<sub>2</sub> emissions worldwide compared to 2010 levels
- By 2050:
  - Limiting Global Warming to 1.5°C Worldwide
  - Net Zero CO<sub>2</sub> emissions

## Methodology

## 方法論

### 1. Software Operations

- Data Sources: EcoMeow will use Python scripts to scrape and clean data from sources like Worldometer, focusing on CO<sub>2</sub> levels, energy consumption, population growth, deforestation rates, and other environmental indicators.
- The models are deployed on a VPS with 4 cores and 16 GB of RAM, running continuously to capture new, up-to-date over a 24 hour period.
- Containerization ensures a modular, scalable, and efficient software stack, with real-time data analysis and predictive modeling displayed for future planning (e.g. 2030, 2050).



### 2. Character Design

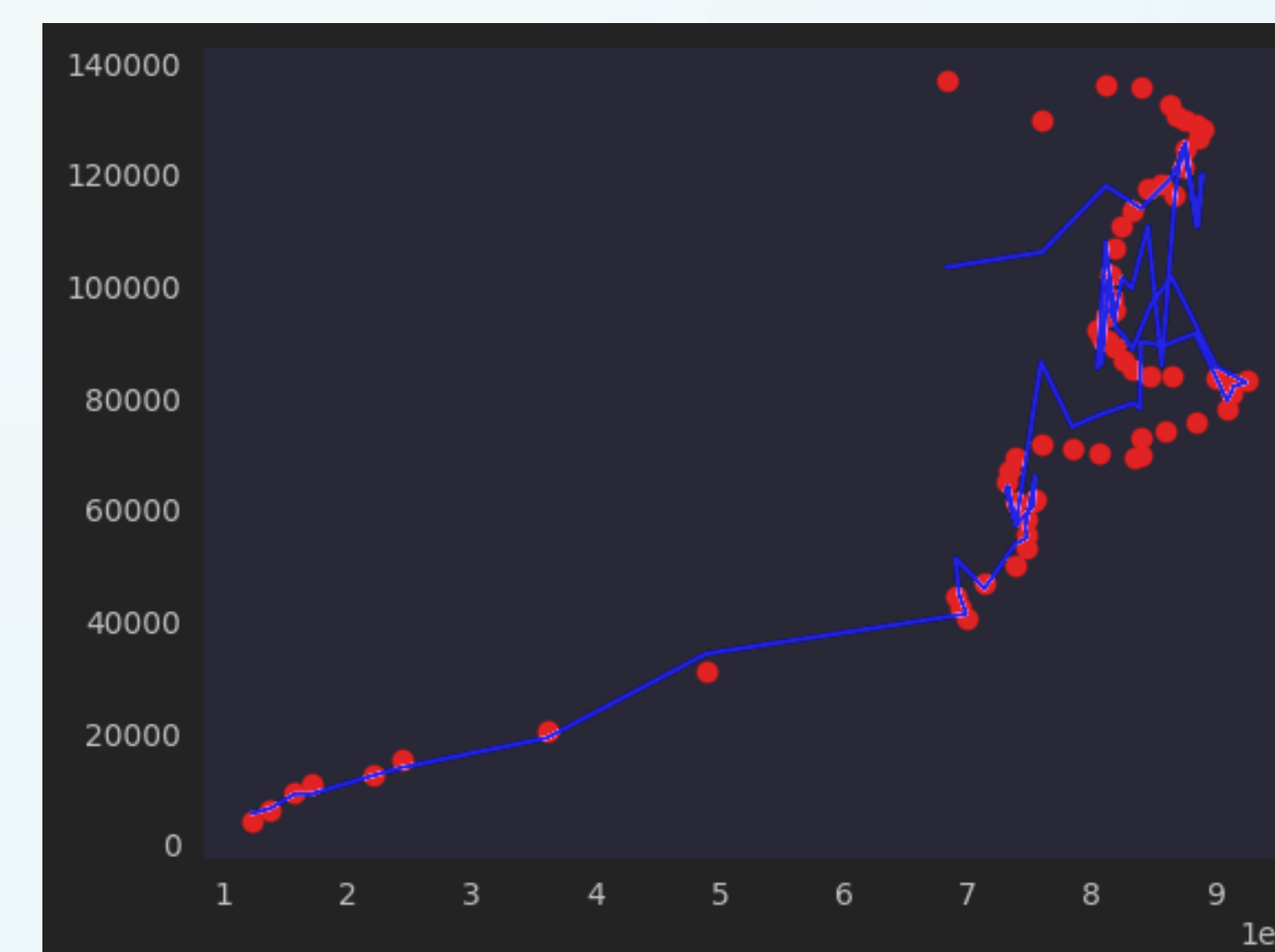
- Eko Eko Chan: A user-friendly cat mascot designed to engage the public and make the software more accessible.
- UX/UI Design: The design focuses on creating a seamless user experience, with intuitive interfaces and visually appealing elements.
- User Interaction: The mascot's role in interacting with the data and guiding users through the platform enhances the system's educational impact and ease of use.

## Methodology

## 方法論

### 3. Machine Learning Predictions

- The software will employ Polynomial Regression and Random Forest Regression, both chosen for their runtime computational efficiency and ability to balance underfitting and overfitting, making them suitable for predicting CO<sub>2</sub> emissions and energy consumption.
- For each type of data we chose to use a suitable type of Regression algorithm, depending on the complexity of the data we expect to receive. For example, for population data that simply grows, we can use Polynomial Regression. But for data that is more complex in nature, like Total CO<sub>2</sub> Emissions or Global warming, we will use Random Forest Regression instead for its accuracy for more complex patterns.
- The models we used all have a self-reported accuracy of over 90%.\*



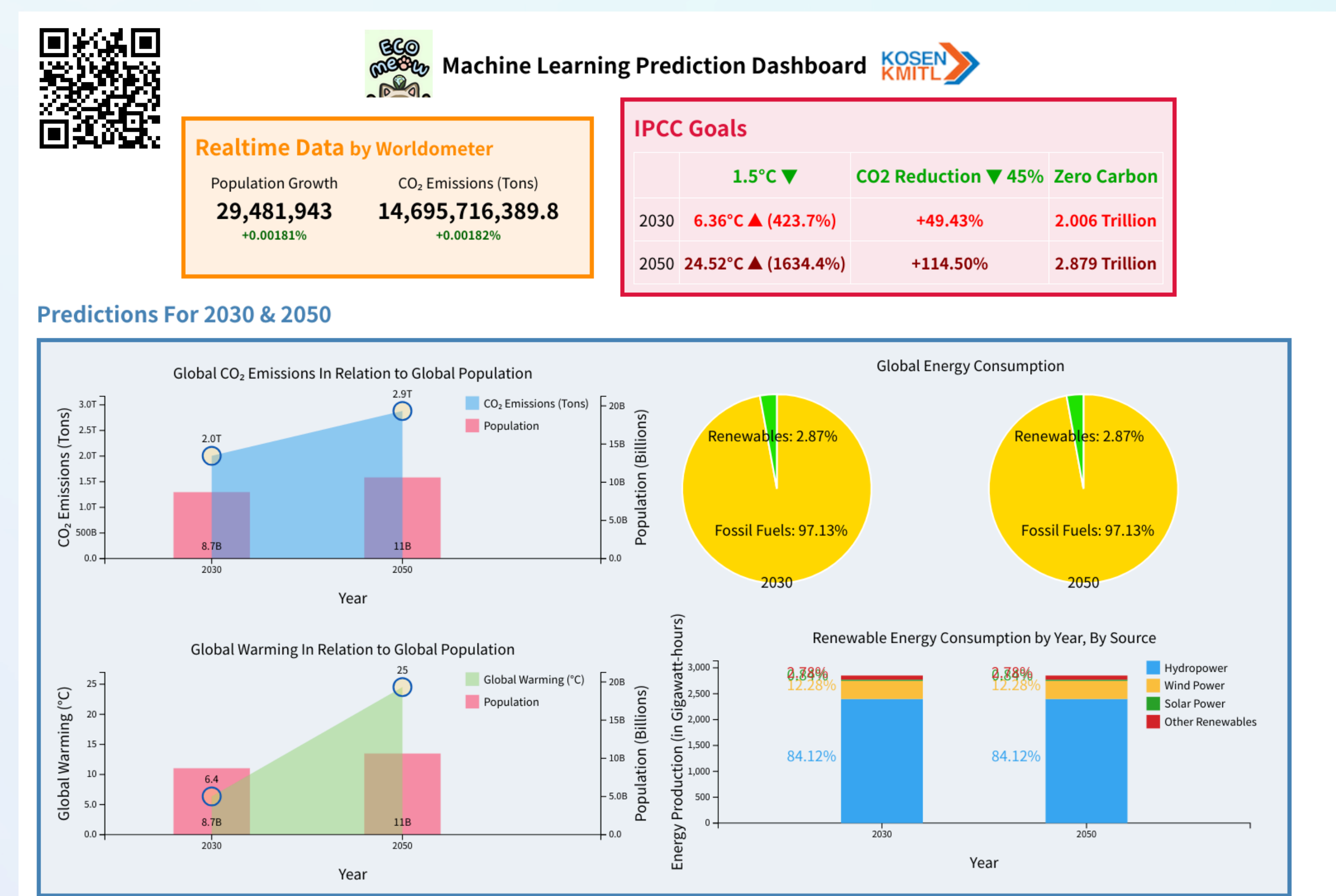
◀ Prediction Example: Using Random Forest Regressor.

▼ Training results

```
Train set evaluation:
MAE: 6581.647398206845
MSE: 94719394.45305389
RMSE: 9732.38893864471
R2 Square 0.9248188392923131
```

## Dashboard

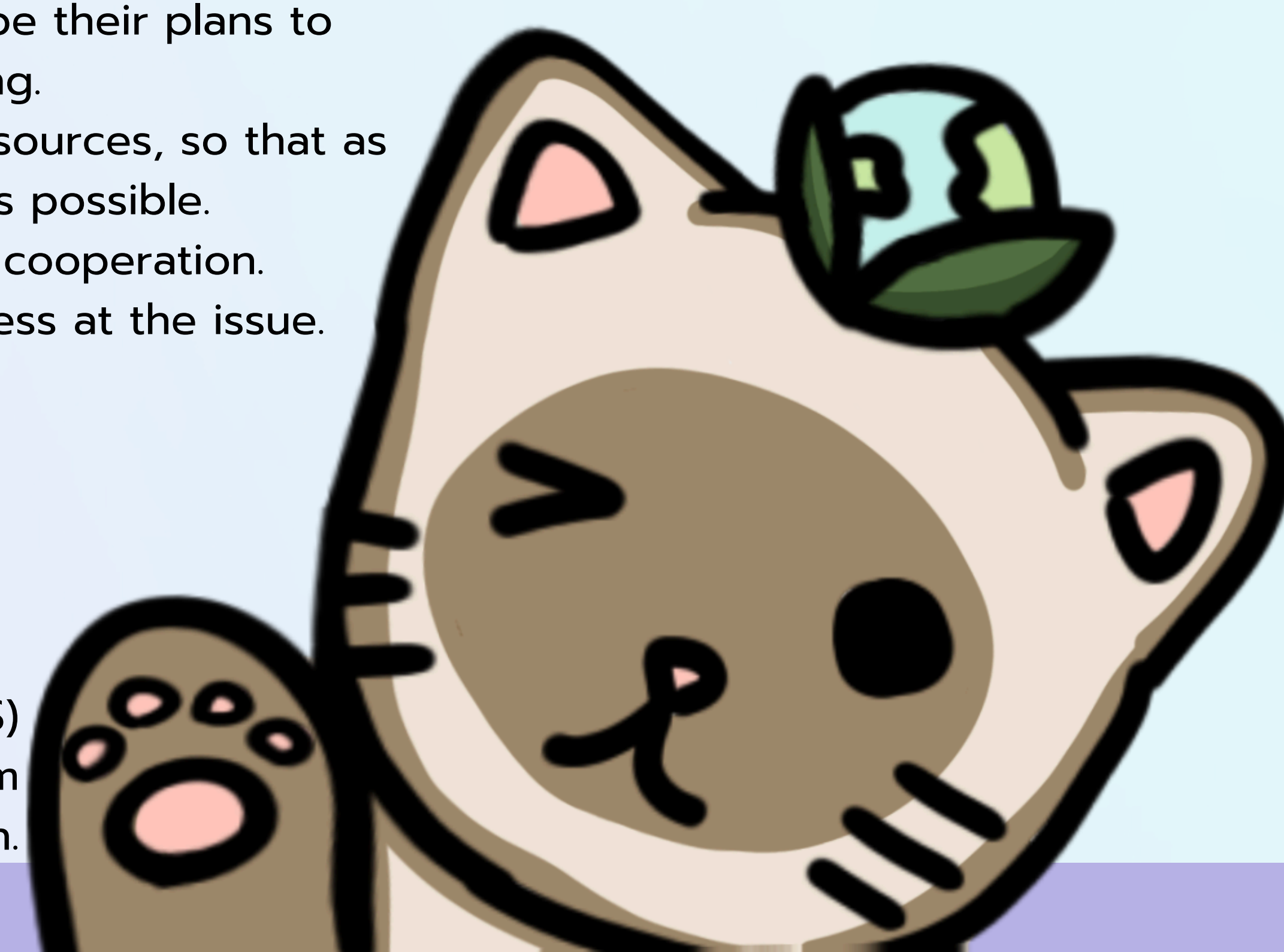
## ダッシュボード



## Advantages

## 長所

- Companies and Governments can use this data to help shape their plans to reduce global warming.
- Helps in allocating resources, so that as little goes to waste as possible.
- Helps support global cooperation.
- Raises public awareness at the issue.



#### Sources

- Worldometer - <https://www.worldometers.info/>
- Our World in Data
  - <https://ourworldindata.org/co2-emissions>
  - <https://ourworldindata.org/grapher/population>
- IPCC - <https://www.ipcc.ch/sr15/>
- UN Net Zero Coalition - <https://www.un.org/en/climatechange/net-zero-coalition>

\*the accuracy values are from the Root Mean Square (RMS) values of the models, both calculated manually and from the sklearn.metrics.r2\_score function.