

# Autonomous Greenhouse Control

An RL approach to controlling and monitoring greenhouses by Team Parshuram. (Kaushal, Arpit, Vyoma, Pathik)



# AGC & Monitoring

Autonomous Greenhouse Control & Monitoring is a reinforcement learning powered solution to manage greenhouses, increase their efficiency and monitor their environment.





01

...

## INTRODUCTION

What is AGC? Why  
AGC? How AGC?

02

...

## RESEARCH AND DEVELOPMENT WORK

03

...

## CREATING A SOLUTION AND BUILDING A PRODUCTION READY PIPELINE

04

...

## PRODUCT DEMO

# 01

...

## INTRODUCTION

A small action for a robot, a  
huge leap for plants? 🧑🌾



# What is AGC?

Autonomous  
Greenhouse Control  
is a research area  
where we find  
different ways to  
control greenhouses  
autonomously

## How AGC?

Using predetermined  
algorithms, machine  
learning approaches  
& reinforcement  
learning.

## Why AGC?

- Reduces load of growers
- Increase produce quality and quantity




# 02

...

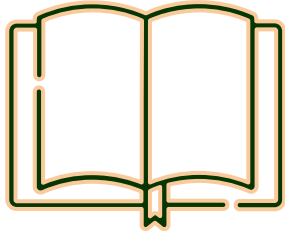
## RESEARCH AND DEVELOPMENT WORK

Can robots grow plants?

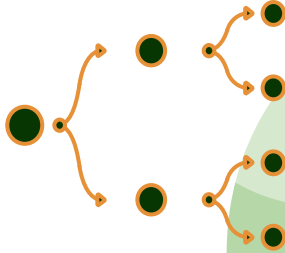
Robot Farmers? 



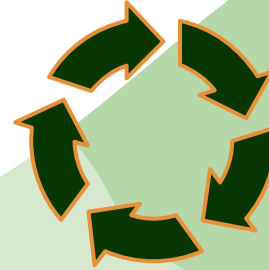
# Our Research And Development



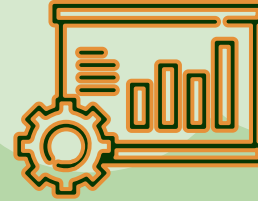
Studying AGC 2 Datasets



Creating Environment and  
Training DQN Agent



Formulating problem as an RL problem



Monitoring Metrics and  
Rewards and Selecting Best  
Models

# Dataset

```
obs[0].head(10)
```

%time	AssimLight	BlackScr	CO2air	Cum_irr	EC_drain_PC	EnScr	HumDef	PipeGrow	PipeLow	...	Acid	%Juice	Bite	Weight	DMC_fruit	Stem_elong	Stem_thick	Cum_trusses	stem_dens
.000000	0.0	35.0	474.0	0.2	1.9	96.0	7.1	71.000000	58.400002	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.003906	0.0	85.0	470.0	0.3	1.9	96.0	7.0	55.900002	57.500000	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.007812	0.0	96.0	482.0	0.3	1.9	96.0	7.1	53.099998	62.299999	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.011719	0.0	96.0	472.0	0.3	1.9	96.0	6.8	70.000000	60.099998	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.015625	0.0	96.0	469.0	0.3	1.9	96.0	6.7	69.300003	57.000000	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.015625	0.0	96.0	479.0	0.3	1.9	96.0	7.4	57.000000	54.200001	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.019531	0.0	96.0	487.0	0.3	1.9	96.0	6.5	48.299999	51.900002	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.023438	0.0	96.0	485.0	0.3	1.9	96.0	6.9	59.500000	49.900002	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.027344	0.0	96.0	483.0	0.3	1.9	96.0	7.3	69.400002	47.900002	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6
.031250	0.0	96.0	476.0	0.3	1.9	96.0	6.6	63.400002	46.599998	...	13.3	68.0	193.0	9.5	8.77	15.0	8.0	1.0	2.6

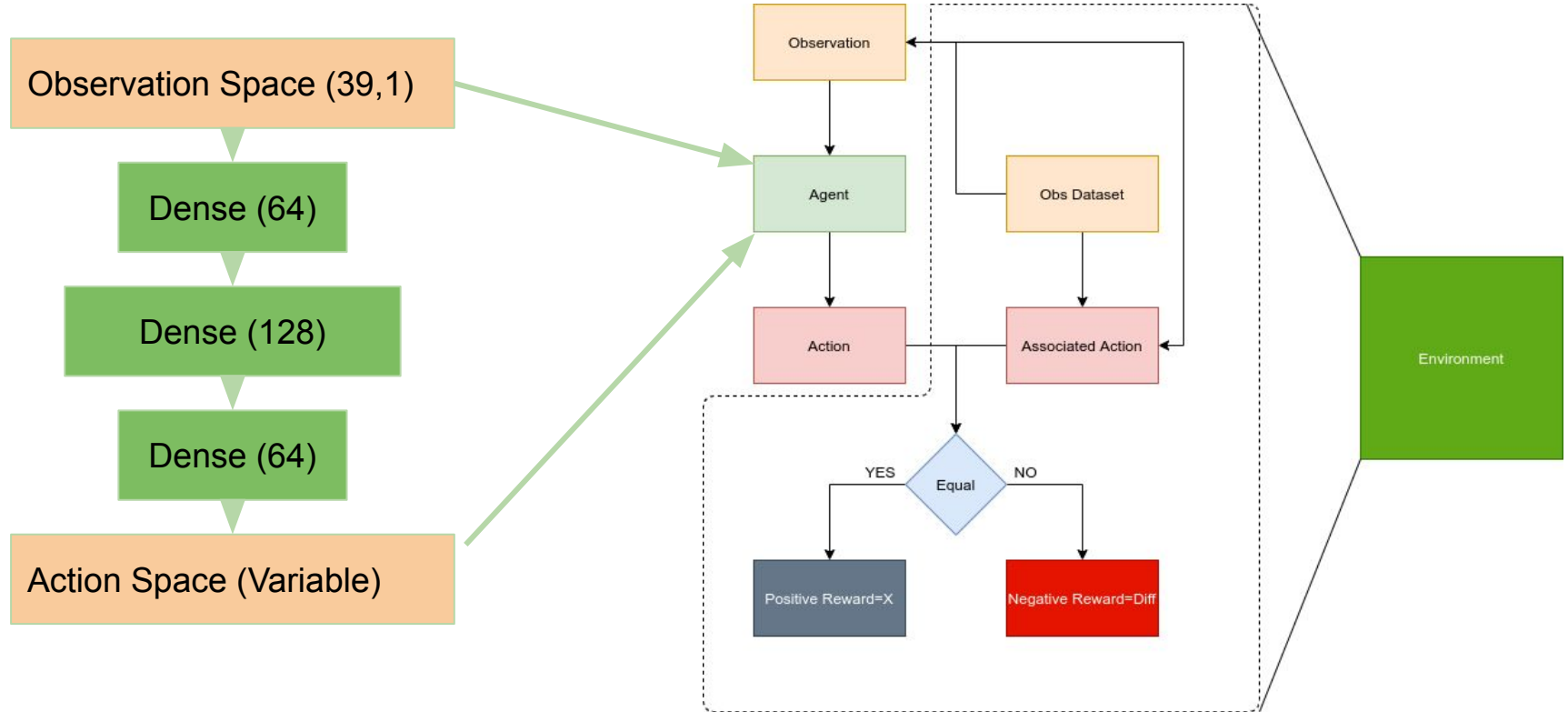
39 columns

```
actions[0].head(10)
```

	assim_sp	co2_sp	dx_sp	int_blue_sp	int_farred_sp	int_red_sp	int_white_sp	scr_blk_sp	scr_enrg_sp	t_grow_min_sp	t_heat_sp	t_rail_min_sp	t_vent_sp	water_sup_intervals_sp_min	window_
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	700.0	0.0	0.0	0.0	0.0	0.0	0.0		0.0

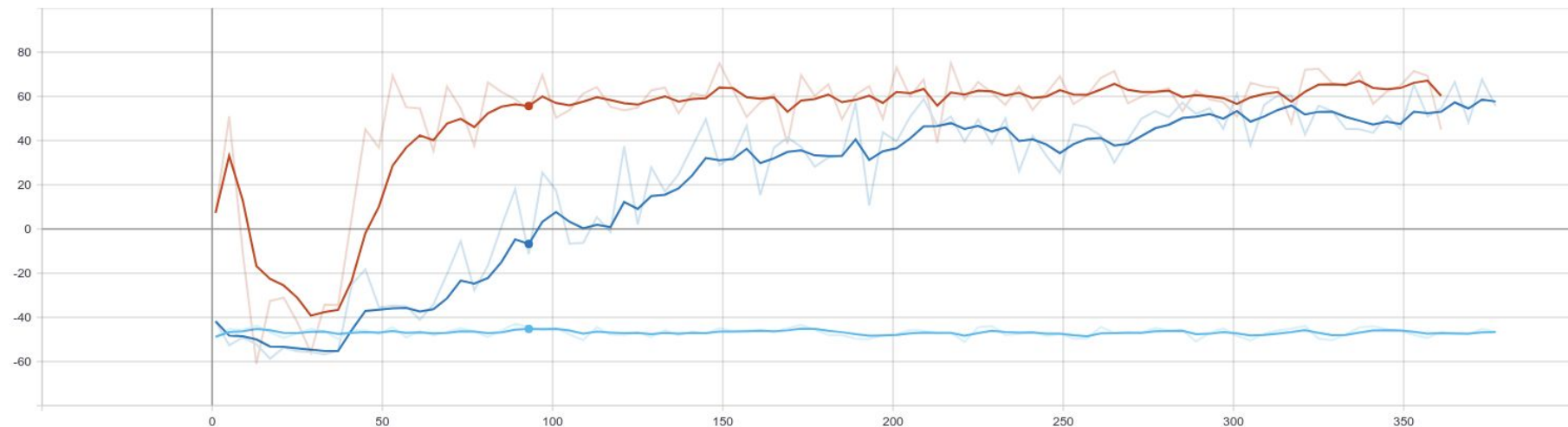


# Our Approach



# Metrics

reward\_avg  
tag: reward\_avg



Name	Smoothed	Value	Step	Time	Relative
AGCRL-RandomPolicy	-45.23	-44.54	93	Wed Apr 14, 11:10:08	7s
AGCRL-Team0_AssimLogs	-6.775	-11.4	93	Wed Apr 14, 09:30:11	8m 47s
AGCRL-Team1(PreTrained)_AssimLogs	55.64	53.93	93	Wed Apr 14, 10:25:56	9m 49s

# 03



## CREATING A SOLUTION AND BUILDING A PRODUCTION READY PIPELINE

I need to pay robots who farm,  
hence I should monitor their  
performance 🤔



# OUR SOLUTIONS

Integration of Dynamic  
Metric Monitoring System  
with Tableau using  
PostgreSQL on AWS RDS

## Tableau Integration



Created an API (using  
FastAPI) that interacts with  
created Environment and  
Agent. Deployed on AWS EC2

## Backend API to interact with Simulation and Agent

Built an Environment and  
Agent Metric Monitoring  
System using Dash Python,  
Plotly deployed on EC2

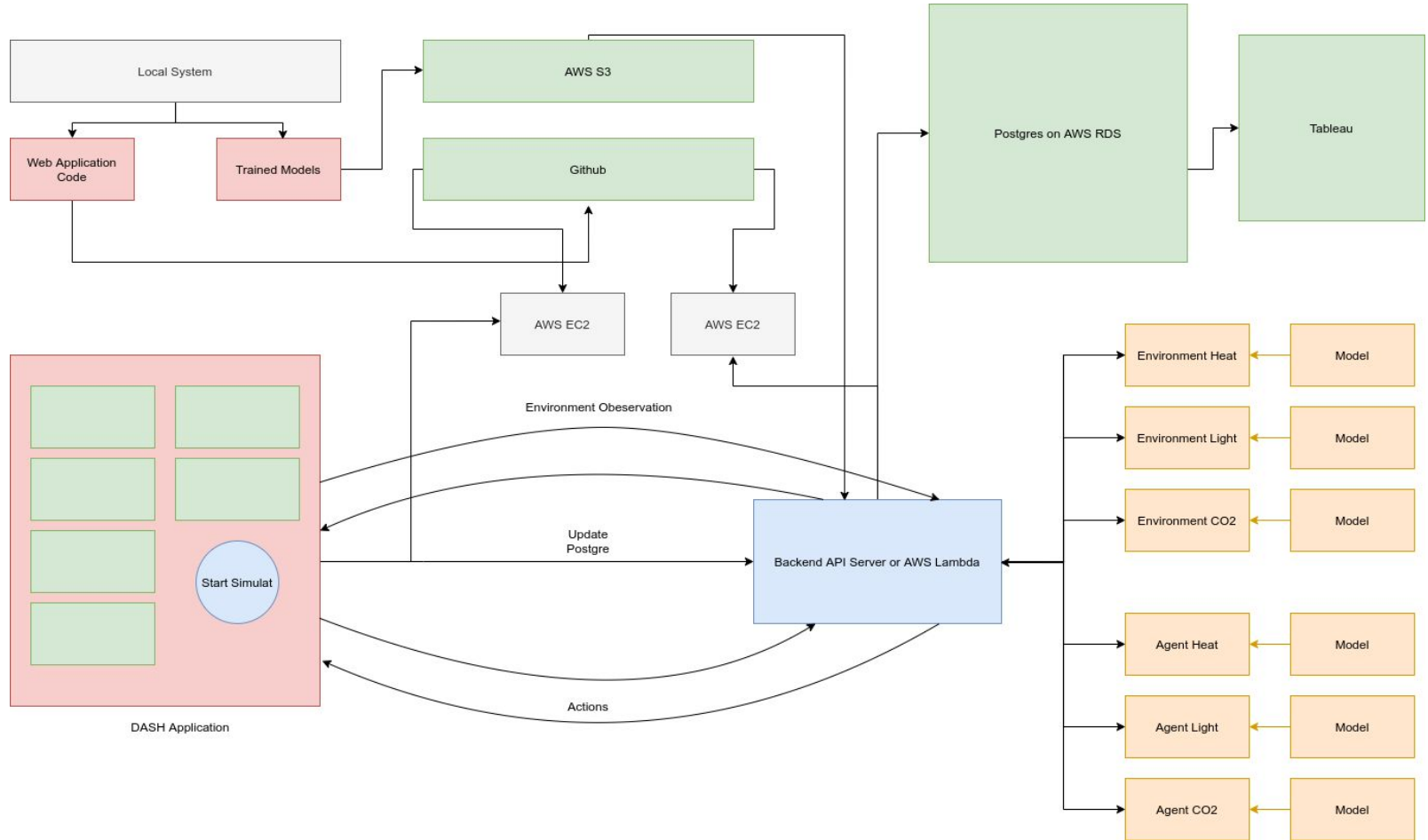
## Environment and Agent Monitoring System



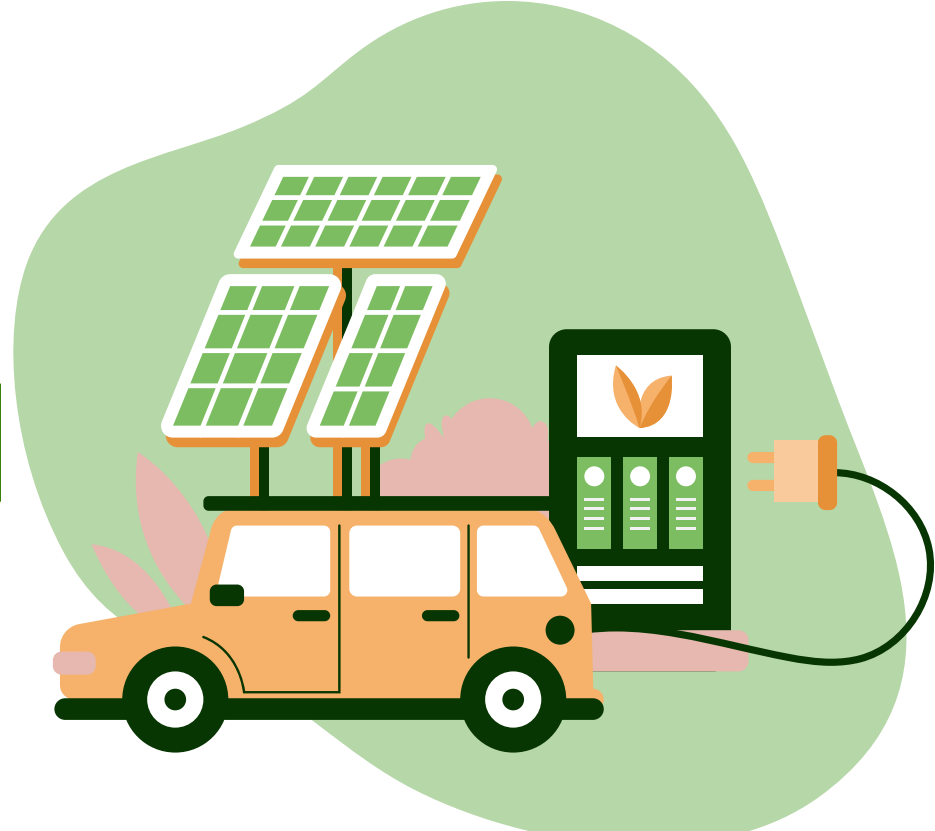
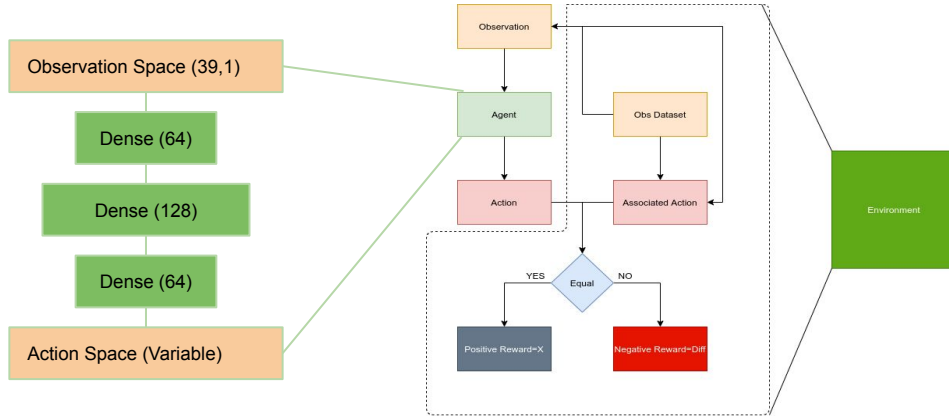
It's the farthest planet from  
the Sun and the fourth-  
largest in our Solar System

## Greenhouse Simulation Environment & Agent

# PIPELINE



# Greenhouse Simulation and Agent



# Backend API

## Root

GET / Read Root

## Light Simulation

POST /light/predict Predict

POST /light/environment Assim Env

GET /light/reset Reset Assim Env

GET /light/reset\_team Reset Assim Env Team

POST /light/teamindex Set Teamindex

GET /light/randomstep Randomstep

GET /light/updatetableau Updatetableau

## Water Simulation

POST /water/predict Predict

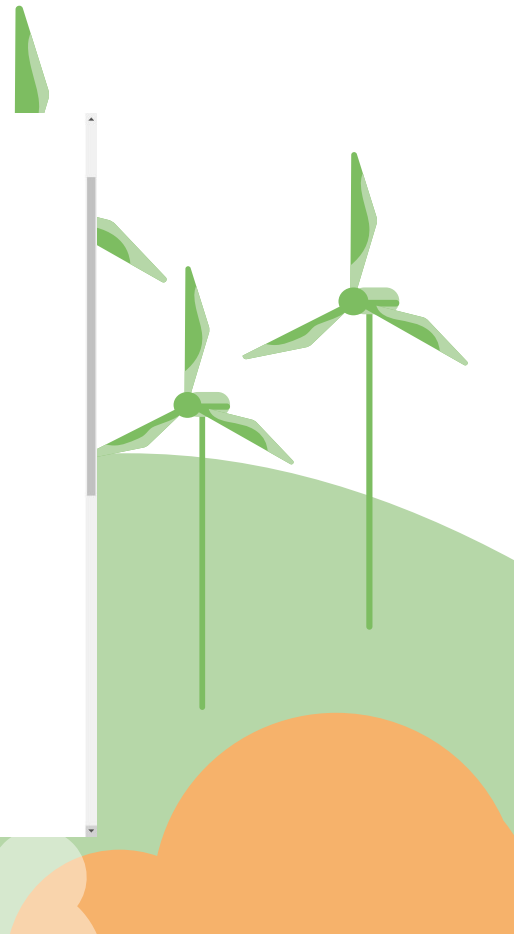
POST /water/environment Water Env

GET /water/reset Reset Water Env

GET /water/reset\_team Reset Water Env Team

POST /water/teamindex Set Teamindex

GET /water/randomstep Randomstep



# Environment and Agent Monitoring System

START SIMULATION

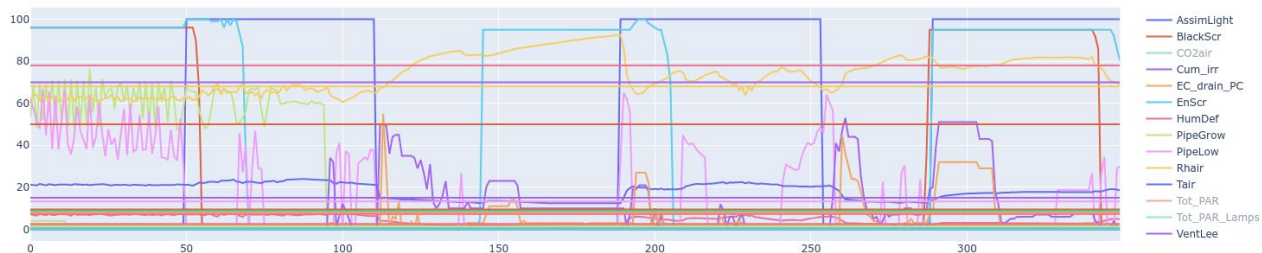
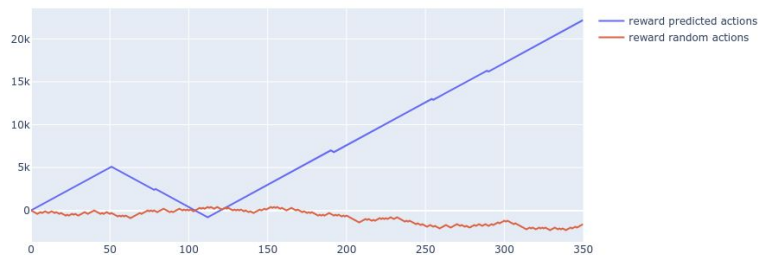
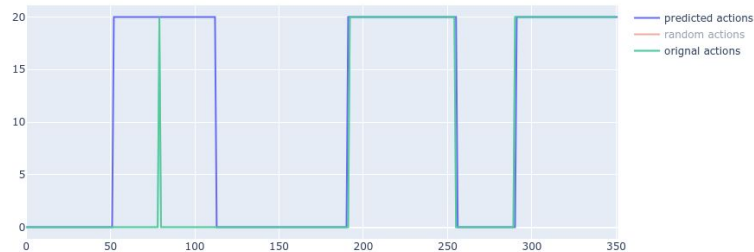
STOP SIMULATION

RESET SIMULATION

RESET TEAM

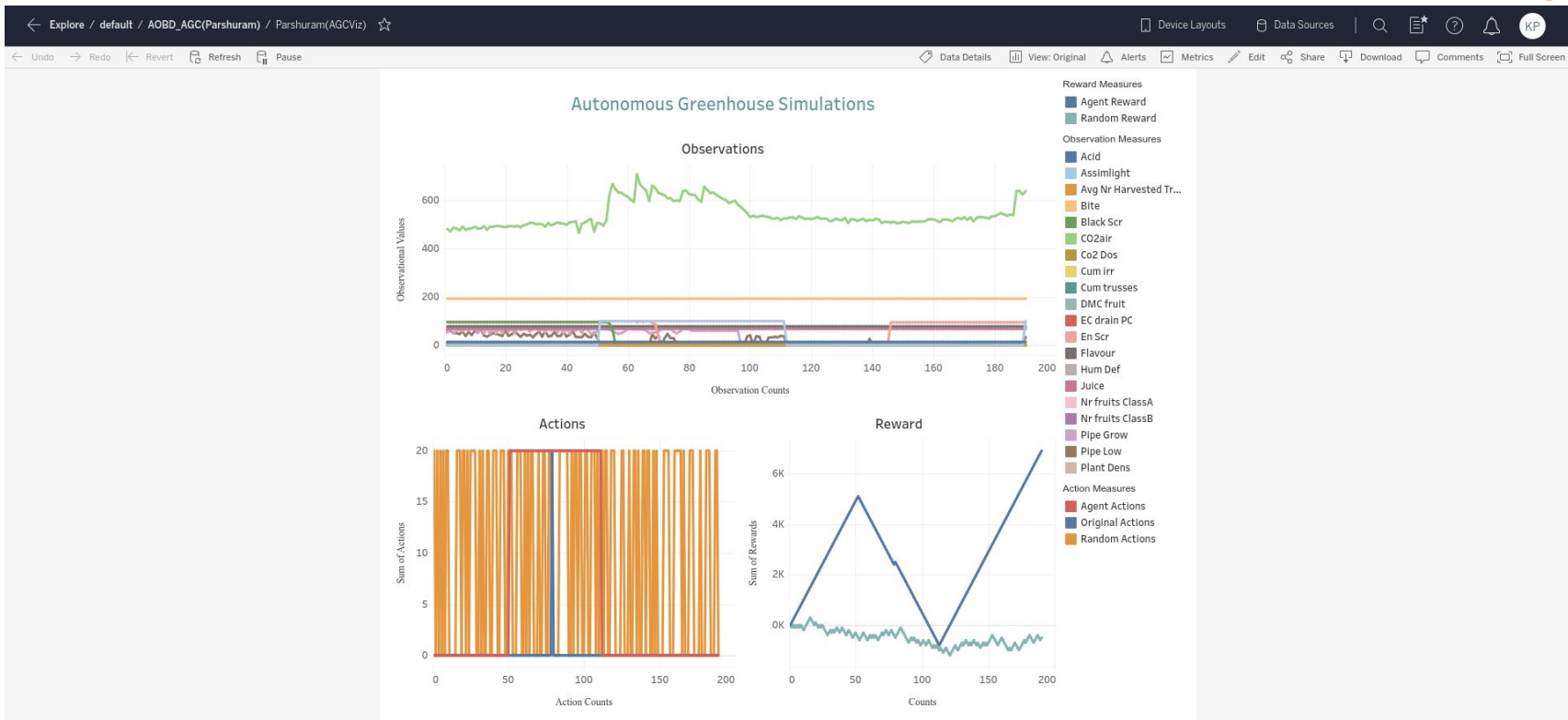
Light

UPDATE TABLEAU





# Tableau Integration




# 04

...

## Product Demo

Now presenting: Something

Cool 





**OUR HOME:**

**<http://www.autogreenhouse.ml/>**



# THIS IS OUR TEAM Parshuram

Kaushal Patil

Arpit Vaghela

Pathik Patel

Vyoma Patel



# Thanks!



Do you have any questions?

