### Team:

Ram, Joel, Srinand

### Topic:

Exoplanet Habitability Classifier

### **Algorithms:**

Decision Trees, FCNN, GAN, GUI, CNN

### Datasets:

<https://www.kaggle.com/datasets/chandrimad31/phl-exoplanet-catalog>

<https://www.kaggle.com/datasets/dandandooo/images>

### Abstract:

With modern issues such as global warming, climate change and depletion of natural resources, combined with an increase in interest in exploring the expanses of space, the need to find a new home for the human race has exponentially increased over the years. Presently, the likelihood that we must abandon Earth for a new home has become far more likely, and in this current climate, finding a habitable exoplanet (planet outside our solar system) is imperative. Due to this circumstance, we have decided to develop an algorithm to take a user's input of data discovered about an exoplanet and determine whether it is habitable. This project can help save the human race as scientists and astrophysicists can quickly determine if a planet is potentially habitable to find a new home for humans quicker.

### Introduction:

This application is designed to predict whether an exoplanet is habitable or not based on input data. We use decision trees to find the columns which best represent the data the most. We then feed them as inputs in a FCNN to classify whether they are habitable (suitable for life) or not. Then we make an AI-generated image of the planet based on the data we had using a GAN.

### Related Works:

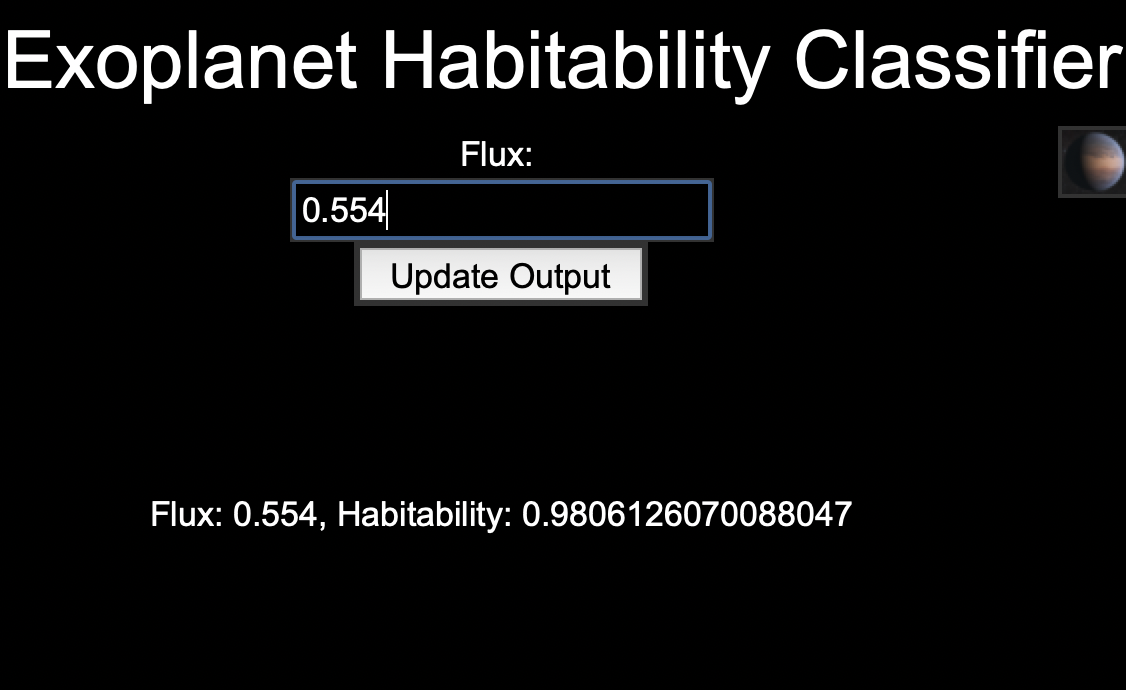
We used concepts we learned in class such as FCNNs, CNNs and Decision Trees. As we didn’t know much python nor any AI/ML information before this course, most of the code is stuff we learnt in this course barring GANs and GUIs, which we needed to learn about in our own time. Therefore, everything we learnt in class was useful for this project as we needed a strong grasp on neural networks, image classification and visual interfaces.

### Methodology:

Image of code breakdown

### Results:

The final product has the user entering all the required parameters that relate to characteristics of the exoplanet, and classifies it to either Habitable or Not Habitable. Taking the characteristics, it also generates an image of what the exoplanet might look like. All of this is combined in one visually appealing window using a GUI, Tkinter.



### Conclusion:

Everyone in our group is pleased with the way our project turned out. From coming to the camp with 0 python knowledge to creating a functional project that can generate entire images of planets, we feel that we have done pretty well. In the future, we would love to improve this by utilizing more columns in the NN to have a more precise output, and possibly making the interface more visually appealing. In addition, we would like to make our project faster by increasing the efficiency.

### Timeline

Decision tree (Joel) -> FCNN (Ram)

-> GUI (Ram)

-> Image Generation (Srinand, Joel)

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Goals Completed:

* Finished cleaning data
* Decision tree
* Neural Network
* Image Generator
* GUI
* GAN
* FCNN
* CNN

### Current Goals:

* None

### Day 1

Plan: decide on what to do for the project

Update: decided on exoplanet habitability predictor/classifier/finder

Update: can’t use cnn or lstm for nn part, fcnn is best

Update: can’t use GAN for image generation as there aren’t enough pictures

Update: calling an AI image generator (craiyon, dalle) could be viable for output section

Update: may have to scrap image generation and create new output as most options require image dataset

Update: finished getting dataset and classifying by habitability

Accomplished: Finished setting up dataset and decision trees

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 2

Plan: Clean the dataset/preprocessing, finish decision tree and start on the neural network.

Updates:

* Finished setting up dataset
* Finished data cleaner
* Finished preprocessing
* Working on decision tree code

Accomplished: Dataset is ready to use and decision tree is almost done

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 3

Plan: Complete decision tree code.

Updates:

* Working on neural network code
* Working on decision tree code
* Decision tree code is done, issues persist that need to be debugged
* Decided on using some kind of chatbot for final output

Accomplished: Decision tree code has main parts done (entropy, information gain, tree builder)

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 4

Plan: Debug decision tree code, specifically runner

Updates:

* Debugging decision tree code.
* Want to look into alternatives to chatbot, looking into implementing API for craiyon AI image generator and use variables as keywords
* Need to rework preprocessing to skip P\_STATUS
* <https://pypi.org/project/craiyon.py/> - implement ai image gen
* Learnt how to implement ai image generator with variables (might use in project output)

Accomplished: decision tree is finished enough to get the best column, now working on fcnn

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 5

Plan: Finish the nn code, start debugging it

Updates:

* Nn code has error as nan, trying to debug
* Found out issue is with y\_pred in binary crossentropy
* Problem with y\_pred is output, which is set by forward prop

Accomplished:

-Finished most of the chatbot

-Finished Neural network, now debugging

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 6

Plan: Debug nn code, find out why error is nan

Updates:

* After debugging, it seems like issue is with forward prop (line 100)
* Issue is with line 27, weights and biases are nan
* Finished base model of the Chatbot
* Fixed Neural Network NAN error

Accomplished:

* Cleaned up decision tree code
* Fixed neural network debugging
* Decision tree can go into Neural Network

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 7

Plan: Finish the Neural Network and fill in the responses for the Chatbot, start on researching GANS

Updates:

* Working on adding a user input classifier to NN.
* Finished NN
* Chatbot is in bad shape, switched to making GUI to display everything together and an image generator made from a GAN and an image dataset

Accomplished:

* Finished NN

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 8

Plan: Improve chatbot, continue working on GAN, see if NN accuracy.

Updates:

* Using accuracy functions to improve NN
* Issue is with train and test accuracy functions
* Finished set up of image generator
* Still working on GAN (working on image generator but running into bugs (images wont plot (images being the normalized images of the planet)))

Accomplished:

* Debugging accuracy function with NN
* Mostly finished image generation

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 9

Plan: There are issues with NN data with nan values, need to be fixed. Image generation will be finished and the discriminator will be started on.

Updates:

* 10:20 - Fixed NN data NaN value issue, working on image generator
* Discriminator = image accuracy finder
* 11:15 - Finished image generator, working on discriminator

Accomplished:

* Image Generator finished
* NN finished

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### 

### Day 10

Plan: Finish image discriminator, finish output

Updates:

* Image discriminator finished, working on debugging output
* output with GUI finished
* Output integrated with neural network so neural network’s output is used in the GUI
* Debugging both discriminator and GUI

Accomplished:

* Image Discriminator mostly finished
* GUI mostly finished

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 11

Plan: Fully finish image generator and GUI

Updates:

* NN output needs denormalization
* Denormalized NN but it still isn't perfect
* Image generation is now generating spherical shapes (improvement, but needs far more epochs / time)
* The GAN actually generates images, not very accurate but it does resemble the training data somewhat. The problem is that gradient collapse occurs after epoch 150, causing no improvements to be made afterwards.
* Update: We fixed the gradient collapse by optimizing the code and adding weight initialization.

Accomplished:

* Most of code is done, now just improving accuracy

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

### Day 12

Plan: Finish everything

Updates:

* All of the code has been integrated, issues with running when im\_gen is combined
* Needed to comment out some code and reorder some code to fix prev issue.
* Pixel resolution increased to make image look more detailed
* Presentation is done, slides have been distributed among group members. Practicing script.
* Tried to use ReLU to improve NN accuracy, failed

Accomplished:

* Project completed

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***