PROJECT REPORT: ExoHab AI

AI-Powered Habitability Analyzer for Exoplanets

Submitted by:

Name: Avula Maneeswara Venkata Sai

Branch: Electronics and Communication Engineering (ECE)

Year: 3rd Year

College: Dhanekula Institute of Engineering and Technology, Ganguru

---

ABSTRACT

The discovery of exoplanets has accelerated rapidly, yet identifying those capable of supporting life remains a challenge due to incomplete astronomical data. ExoHab AI is a full-stack Machine Learning SaaS application designed to solve this problem. Unlike standard black-box models, ExoHab integrates a custom Physics Engine (based on Keplerian laws) with Explainable AI (SHAP) to predict habitability with high precision.

This project addresses the critical issue of data scarcity by imputing missing stellar parameters using astrophysical formulas before feeding them into an XGBoost Classifier. The final system achieves state-of-the-art accuracy and is deployed as a cloud-based web application, offering real-time analysis, 3D visualization, and detailed explainability for astronomers and enthusiasts.

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

Current methods for identifying habitable planets rely heavily on manual calculations or simple "Goldilocks Zone" checks. However, raw data from missions like Kepler and TESS is often incomplete—missing critical values like "Stellar Luminosity" or "Planet Mass." Furthermore, traditional AI models give a binary "Yes/No" result without explaining *why* a planet is habitable.

1.2 Project Objectives

1. **Physics-Informed AI: To build a model that understands astrophysical laws, not just statistical patterns.**
2. **Handling Imbalance: To solve the drastic 99:1 imbalance between non-habitable and habitable planets in the NASA archive.**

3. **Explainability: To implement "Glass Box" AI that visualizes the reasoning behind every prediction.**
4. **Accessibility: To deploy a user-friendly web interface that allows anyone to upload NASA data and get instant results.**

CHAPTER 2: SYSTEM ARCHITECTURE

2.1 High-Level Design

ExoHab utilizes a Monolithic MVC (Model-View-Controller) architecture designed for low-latency inference. The data flow consists of five distinct layers:

1. **Input Layer: Users upload raw NASA Archive CSV files or input single planet parameters via the Web UI.**
2. **Preprocessing Layer (The Physics Engine): A custom module that imputes missing astronomical data using domain-specific laws (see Section 3.1).**
3. **Inference Layer (AI Core): The processed vector is fed into an XGBoost Classifier trained on balanced data.**
4. **Explainability Layer (XAI): A SHAP (Shapley Additive Explanations) TreeExplainer calculates the contribution score of every feature.**
5. **Presentation Layer: Results are rendered via Flask (Jinja2) and interactive JavaScript (Plotly.js).**

2.2 Tech Stack

- **Frontend: HTML5, Bootstrap 5, JavaScript (Plotly.js for 3D mapping).**
- **Backend: Python 3.10, Flask (REST API), Gunicorn Server.**
- **Data Science: Pandas, NumPy, Scikit-Learn, Imbalanced-Learn.**
- **Machine Learning: XGBoost (Gradient Boosting), SHAP.**
- **Deployment: Render Cloud Platform (PaaS).**

CHAPTER 3: METHODOLOGY & IMPLEMENTATION

3.1 The Physics Engine (Data Imputation)

Unlike standard ML pipelines that use mean/median imputation, ExoHab uses Domain-Specific Imputation to ensure scientific accuracy.

- **Kepler's Third Law: Used to derive Orbital Period (P) or Semi-Major Axis (a) if one is missing.**

$$P^2 \propto a^3$$

- **Stefan-Boltzmann Law: Used to calculate Star Luminosity (L) if Radius (R) and Temperature (T) are known.**

$$L = 4\pi R^2 \sigma T^4$$

3.2 Machine Learning Pipeline

- **Data Cleaning: Rigorous feature selection was performed to isolate the top 10 physics-based parameters (e.g., Insolation Flux, Equilibrium Temperature).**
- **Handling Imbalance (SMOTE): The dataset had a 99:1 imbalance (mostly dead planets). I implemented SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic examples of habitable planets, teaching the model to recognize them effectively.**
- **Model Selection: Multiple architectures were benchmarked:**
  - ***Baseline:*** **Logistic Regression.**
  - ***Ensemble:*** **Random Forest + SMOTE.**
  - ***Champion:*** **XGBoost + SMOTE (Selected for achieving 100% Precision on validation data).**

CHAPTER 4: RESULTS & ANALYSIS

4.1 Model Performance

The final SMOTE + XGBoost model demonstrated superior performance compared to baseline models.

- **Precision: 100% (Crucial: effectively eliminated False Positives).**
- **Recall: High sensitivity in detecting "Optimistic" candidates.**
- **ROC-AUC: Demonstrates strong separability between habitable and non-habitable classes.**

4.2 Visual Analytics

The application generates three key visualizations for every dataset:

1. **Feature Importance: A bar chart ranking which factors (e.g., Planet Radius vs. Star Temp) influenced the decision most.**
2. **Correlation Matrix: A heatmap visualizing how different parameters interact.**
3. **3D Galaxy Map: An interactive plot showing the distribution of planets in the "Goldilocks Zone" (Star Temp vs. Planet Radius vs. Equilibrium Temp).**

CHAPTER 5: DEPLOYMENT & API

5.1 Cloud Infrastructure

The application was finalized and deployed to Render to make it publicly accessible.

- **Web Server: Gunicorn (Green Unicorn) WSGI server configured with sync workers.**
- **Environment: Python 3.10+ with dependencies managed via requirements.txt.**
- **Live URL: https://exohabai.onrender.com/**

5.2 API Endpoints

- **POST /predict_single: Analyzes a single planet based on JSON input. Returns a prediction ("Habitable", "Non-Habitable") and a SHAP reasoning list.**
- **POST /predict_bulk: Processes massive CSV files (>5,000 rows) and returns a JSON array of results.**

CHAPTER 6: CONCLUSION & FUTURE SCOPE

6.1 Conclusion

ExoHab successfully bridges the gap between raw astronomical data and actionable insights. By combining a Physics Engine with advanced Machine Learning, the tool allows researchers to identify habitable candidates instantly, without manual calculation.

6.2 Future Roadmap

1. **Spectral Analysis: Integration of Convolutional Neural Networks (CNNs) to analyze spectral light curves for atmospheric composition detection (e.g., Oxygen/Methane signatures).**
2. **Real-Time API Sync: Automating the daily fetching of new confirmed planets from the NASA Exoplanet Archive API.**
3. **Procedural Generation: Using Three.js to render realistic 3D textures of planets based on their predicted temperature and composition.**

REFERENCES

1. **NASA Exoplanet Archive (Caltech).**
2. **Planetary Habitability Laboratory (PHL) - UPR Arecibo.**
3. ***Scikit-Learn & XGBoost Documentation.***
4. ***Explainable AI (SHAP) - Lundberg & Lee (2017).***