Predicting the Habitability of Exoplanets Using Machine Learning

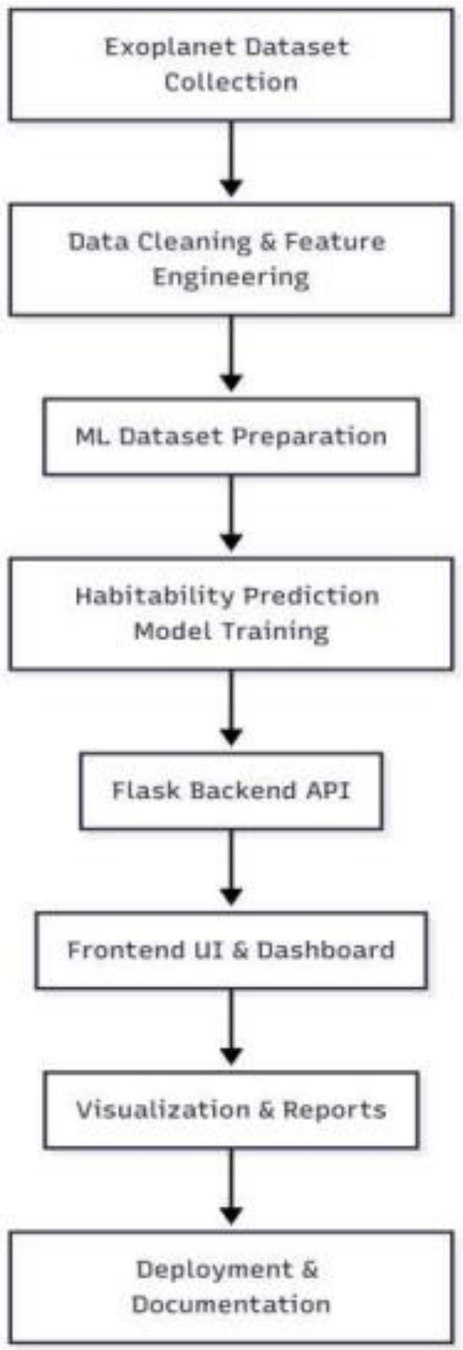
Project Statement:

The discovery of exoplanets has accelerated in recent years, but identifying which planets could potentially support life remains a challenge. Current astronomical methods are limited by observational constraints, massive datasets, and complex planetary parameters. This project, ExoHabitAI, aims to use machine learning to predict the habitability of exoplanets based on physical, orbital, and stellar features. The system evaluates exoplanetary characteristics such as radius, mass, orbital period, equilibrium temperature, and host star parameters to classify planets as potentially habitable or not. By leveraging data-driven models, astronomers and researchers can prioritize candidate exoplanets for further study, enhancing the efficiency of observational campaigns.

Modules to be implemented

- Data Collection & Management (Exoplanet Dataset)
- Data Cleaning & Feature Engineering
- Machine Learning Dataset Preparation
- AI Model for Habitability Prediction (ML-Based)
- Backend API Integration (Flask)
- Frontend UI (Bootstrap + HTML + JavaScript)
- Visualization & Dashboard (Habitability Insights)
- Deployment & Documentation

Workflow for Predicting Exoplanet Habitability Using Machine Learning


Exoplanet Dataset Collection
Data Cleaning & Feature Engineering
ML Dataset Preparation
Habitability Prediction Model Training
Flask Backend API
Frontend UI & Dashboard
Visualization & Reports
Deployment & Documentation

1. Input from CSV File

The dataset was imported from a CSV file named: PS_2026.02.09_09.53.13.csv The file contains 746 rows and 92 columns. Each row represents a confirmed exoplanet, and each column represents a planetary, stellar, or discovery-related feature. The dataset was loaded using pandas with comment lines ignored: pd.read_csv('PS_2026.02.09_09.53.13.csv', comment='#') This ensures metadata lines starting with '#' are excluded and only structured tabular data is processed.

CSV Loading Code:

```
import pandas as pd

df = pd.read_csv("PS_2026.02.09_09.53.13.csv", comment="#")
print(df.shape)
print(df.head())
```

2. Data Description

The dataset contains information about confirmed exoplanets (planets outside our solar system).
Each row represents one confirmed exoplanet, and each column represents a specific property related to:

- Planet characteristics
- Host star properties
- Orbital parameters
- Discovery information

This dataset is structured and numerical in nature, making it suitable for machine learning regression tasks.

Dataset Size:

- Total Rows: ~746 records
- Total Columns: ~90+ features

Types of Features:

A) Planetary Features

The dataset contains the following categories of features:

- `pl_orbper` → Orbital period
- `pl_orbsmax` → Orbital semi-major axis
- `pl_rade` → Planet radius (Earth units)
- `pl_bmasse` → Planet mass (Earth units)
- `pl_eqt` → Planet equilibrium temperature

These features help describe the physical nature of the planet.

B) Stellar (Host Star) Features

These describe the star around which the planet orbits:

- `st_teff` → Effective temperature of star
- `st_mass` → Stellar mass
- `st_rad` → Stellar radius
- `st_met` → Metallicity
- `st_logg` → Surface gravity

These features influence planetary formation and detection probability.

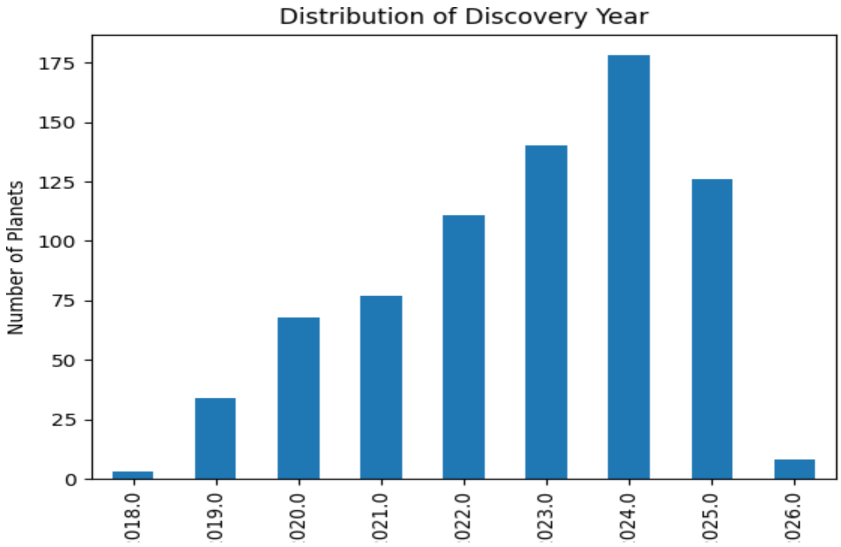
C) Discovery Information

These include details about how and when the planet was discovered:

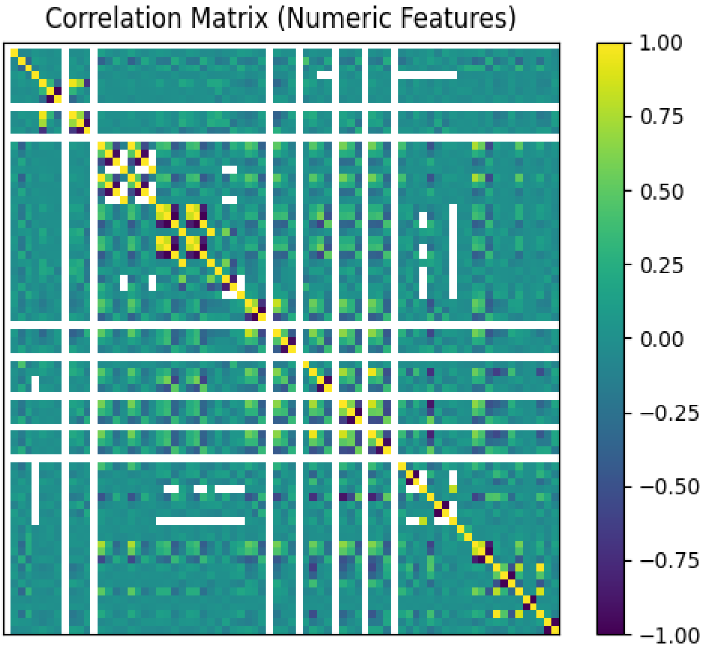
`disc_year` → Discovery year
`discoverymethod` → Detection method
`disc_facility` → Telescope or facility use

This is useful for time-based analysis and trend studies.

3. Discovery Year Distribution Graph



4. Correlation Matrix Graph



5. Data Preprocessing Theory

Data preprocessing prepares raw data for machine learning models. Steps applied in this project include: 1. Removing duplicate records to avoid biased learning. 2. Handling missing values in numerical columns using median imputation. 3. Filling categorical missing values with 'Unknown'. 4. Encoding categorical features into numeric form. 5. Splitting dataset into training and testing sets.