

ExoHabitAI Project Documentation

1. Project Overview

The objective of the ExoHabitAI project is to predict the habitability of exoplanets using astrophysical and stellar features. The project follows a complete end-to-end machine learning pipeline, starting from raw data cleaning and preprocessing, through feature engineering, and finally model training, evaluation, and saving outputs.

2. Data Source

- Dataset: Exoplanet data (CSV format)
- Source: NASA Exoplanet Archive–style structured dataset
- Initial size: ~300K+ rows, ~280+ columns

The dataset contains planetary, orbital, and stellar parameters such as planet radius, temperature, orbital period, stellar mass, and luminosity.

3. Phase 1 – Data Cleaning & Preprocessing

➤ Initial Inspection

- Loaded the raw dataset using Pandas
- Checked:
 - Dataset shape
 - Column names
 - Data types
 - Duplicate rows
 - Missing values

➤ Column Selection

Selected only scientifically meaningful columns relevant to habitability analysis:

- Planet features: radius, orbital period, equilibrium temperature, semi-major axis
- Stellar features: temperature, radius, mass, luminosity
- Metadata: discovery year, discovery method

This reduced noise and unnecessary columns.

➤ Handling Missing Values

- Columns with more than **70% missing values** were dropped
- Numeric columns: filled using **median imputation**
- Categorical columns: filled using **mode**

➤ **Duplicate Removal**

- Duplicate rows were identified and removed

➤ **Physical Validity Checks**

Removed physically impossible values: - Planet radius > 0 - Orbital period > 0 - Equilibrium temperature > 0

➤ **Outlier Treatment**

- Used **IQR (Interquartile Range)** method
- Applied only to critical numerical features

➤ **Phase 1 Output**

- Cleaned dataset saved as:
 - data/processed/preprocessed.csv
- Shape after cleaning: **(33695, 13)**

4. Phase 2 – Feature Engineering

➤ **Planet Mass Analysis**

- Identified all available planet mass-related columns
- Observed large missingness (expected in astronomy datasets)
- Planet mass was not forcibly imputed to avoid scientific bias

➤ **Planet Density**

- Used existing density values where available
- Density was retained with NaNs (handled later during ML preprocessing)

➤ **Stellar Luminosity**

- Confirmed availability of stellar luminosity (`st_lum`)
- Cleaned and standardized for further use

➤ **Phase 2 Output**

- Feature-engineered dataset saved as:
 - data/processed/preprocessed_phase2.csv

5. Phase 3 – Advanced Feature Engineering

➤ **Habitability Index**

Created a composite index combining: - Temperature proximity to habitable range - Planet radius similarity to Earth - Distance from host star - Stellar luminosity

This index converts physical rules into ML-readable numeric signals.

➤ **Stellar Compatibility Index**

Measured how suitable the host star is for life using: - Stellar temperature - Stellar radius - Stellar luminosity stability

➤ **Orbital Stability Factor**

Calculated using: - Orbital period - Semi-major axis

Stable orbits increase long-term habitability potential.

➤ **Categorical Encoding**

- Encoded stellar types and discovery methods using **One-Hot Encoding**

➤ **Feature Scaling**

- Applied **StandardScaler** to normalize numerical features

➤ **Target Variable Creation**

- Created binary target variable: habitable
 - 1 → Potentially habitable
 - 0 → Not habitable

➤ **Phase 3 Output**

- Final ML-ready dataset saved as:
 - data/processed/preprocessed_phase3.csv
- Final shape: **(115617, 40)**

6. Phase 4 – Machine Learning Model Training

➤ **ML Problem Definition**

- Type: Supervised Learning
- Task: Binary Classification
- Target: habitable

➤ **ML-Specific Preprocessing**

Performed in the ML notebook only:

- Removed identifier columns (planet names, host names)
- Selected numeric features only
- Handled remaining NaNs using **median imputation**
- Train-test split (80% train, 20% test, stratified)

7. Models Trained

➤ **Logistic Regression (Final Model)**

- Chosen for simplicity and interpretability
- Regularized, linear classifier

Performance:

- Accuracy $\approx 99.38\%$
- Precision $\approx 99.89\%$
- Recall $\approx 98.87\%$
- F1-score $\approx 99.38\%$
- ROC-AUC ≈ 0.9999

➤ **Random Forest (Comparison Model)**

- Achieved perfect scores (1.0)
- Determined to overfit due to deterministic labels

➤ **XGBoost (Comparison Model)**

- Also achieved near-perfect scores
- Confirmed strong rule-learning capability

8. Overfitting Analysis

- Habitability labels were generated using **deterministic physical rules**
- Tree-based models reconstructed these rules exactly
- Logistic Regression chosen as final model due to:
 - Better generalization
 - Interpretability
 - Reduced overfitting risk

9. Model Saving

- Final trained Logistic Regression model saved using joblib
- Location:
 - `models/logistic_regression_final.pkl`

10. ML Model Output

- Model predictions saved as CSV
- Includes:
 - Actual label
 - Predicted label

- Prediction probability

File: - data/processed/ml_model_output.csv

11. Final Project Structure

```
ExoHabitAI/
  └── data/
    ├── raw/
    └── processed/
      ├── preprocessed.csv
      ├── preprocessed_phase2.csv
      ├── preprocessed_phase3.csv
      └── ml_model_output.csv
  └── notebooks/
    ├── phase1_preprocessing.ipynb
    ├── phase2_feature_engineering.ipynb
    ├── phase3_advanced_features.ipynb
    └── phase4_ml_model_training.ipynb
  └── models/
    └── logistic_regression_final.pkl
  └── README.md
```

12. Conclusion

The ExoHabitAI project demonstrates a complete and scientifically grounded machine learning workflow. The project emphasizes correct data preprocessing, meaningful feature engineering, responsible model selection, and proper interpretation of results. Logistic Regression was selected as the final model due to its interpretability and generalization capability, while advanced models were used for comparison and validation.