Data Preprocessing and Machine Learning Model Documentation

1. Introduction

The number of discovered exoplanets has increased rapidly due to advances in astronomical observation. However, identifying which of these planets could potentially support life is challenging because habitability depends on multiple interacting planetary and stellar factors.

The goal of the **ExoHabitAI** project is to build a supervised machine learning model that predicts the habitability potential of exoplanets using cleaned and processed planetary and stellar data. This document explains the data preprocessing steps and machine learning methodology used in the project in a clear and human-understandable manner.

---

2. Dataset Overview

The dataset contains physical and orbital information about exoplanets and their host stars collected from astronomical databases.

Main Features Used

Planetary Features

- Planet radius (pl_rade)
- Planet mass (pl_bmasse)
- Orbital period (pl_orbper)
- Orbital distance / semi-major axis (pl_orbsmax)
- Equilibrium temperature (pl_eqt)
- Planet density (pl_dens)

Stellar Features

- Stellar effective temperature (st_teff)
- Stellar luminosity (st_lum)
- Stellar metallicity (st_met)
- Stellar spectral type (st_spectype)

The raw dataset contains missing values, duplicates, and extreme values, making preprocessing necessary before applying machine learning models.

---

3. Data Preprocessing

3.1 Feature Selection

Only scientifically relevant planetary and stellar features were selected for model training. Metadata, identifiers, and unrelated attributes were removed to reduce noise and improve model generalization.

3.2 Handling Missing Values

Missing values were handled using statistically sound methods:

- Numerical features were filled using the **median**, which is resistant to outliers.
- Categorical features (stellar spectral type) were filled using the **mode**.
- Completely empty rows were removed.

This approach preserves the natural distribution of the data while minimizing bias.

3.3 Duplicate Removal and Outlier Treatment

Duplicate records were removed to prevent data leakage.

Outliers were handled using:

- **Z-score filtering** to remove extreme anomalies
- **IQR-based clipping** to cap unrealistic values

This improved model stability without discarding meaningful data.

3.4 Categorical Encoding and Feature Scaling

Stellar spectral types were converted into numerical form using **One-Hot Encoding**.

Because numerical features have different scales, **StandardScaler** was applied within machine learning pipelines. Scaling was done after train–test splitting to prevent data leakage.

The final cleaned dataset was saved as:

data/processed/preprocessed.csv

---

4. Target Variable Creation

The dataset does not include a direct habitability label. Therefore, a **proxy binary target variable** was created using astrophysical heuristics.

An exoplanet is labeled as:

- **1 (Potentially Habitable)** if it satisfies Earth-like conditions (reasonable radius, sufficient density, and suitable host star temperature)
- **0 (Non-Habitable)** otherwise

This allows supervised learning while maintaining scientific plausibility.

---

5. Machine Learning Methodology

5.1 Problem Formulation

- Learning Type: **Supervised Machine Learning**
- Task: **Binary Classification**
- Input: Cleaned and encoded planetary and stellar features
- Output: Habitability class and habitability probability score

---

5.2 Train–Test Split

The dataset was split into:

- 80% training data
- 20% testing data

A fixed random state was used for reproducibility, and stratified sampling preserved class balance.

---

6. Model Training

6.1 Baseline Models

Baseline models were trained to establish reference performance:

- Logistic Regression
- Shallow Decision Tree

These models provided initial benchmarks for accuracy, recall, and F1-score.

6.2 Advanced Models

To capture non-linear relationships, ensemble models were used:

- **Random Forest Classifier**
- **XGBoost Classifier**

Scikit-learn pipelines were used to combine scaling and model training, ensuring clean and reproducible experiments.

---

7. Model Evaluation and Hyperparameter Tuning

Models were evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- ROC–AUC

Visual evaluation included confusion matrices and ROC curves.

Hyperparameter tuning was applied to the XGBoost model using **RandomizedSearchCV**, tuning parameters such as:

- Number of trees
- Maximum depth
- Learning rate
- Subsampling ratios

---

8. Model Selection and Ranking

The final model was selected based on:

- High recall for the habitable class
- Strong F1-score

- Stable performance on test data

XGBoost was chosen as the final model because it best aligns with the scientific objective of minimizing false negatives.

9. Conclusion

This project demonstrates a complete machine learning pipeline for exoplanet habitability prediction. Through careful preprocessing, robust model training, and scientifically motivated evaluation, the ExoHabitAI system successfully identifies potentially habitable exoplanets.

The final model prioritizes recall for habitable planets, making it suitable for real-world astronomical research and future exploration.