Milestone–2

Data Preprocessing and Machine Learning Model Training for ExoHabitAI

1. Data Preprocessing

1.1 Dataset Loading

The raw dataset was loaded using the Pandas library. Special parameters were used to ensure robustness:

- Commented lines were ignored
- Corrupted or malformed rows were skipped

This ensured successful ingestion of real-world astronomical data without execution errors.

1.2 Initial Data Exploration

The following checks were performed to assess data quality:

- Dataset dimensions were examined
- Missing values were identified column-wise
- Duplicate rows were detected
- A missing value heatmap was generated using Seaborn

These steps provided a clear understanding of inconsistencies and data gaps.

1.3 Feature Selection

The original dataset contained approximately 289 columns.
Only scientifically relevant planetary and stellar attributes were retained:

> Planet radius, Planet mass, Orbital period, Semi-major axis, Equilibrium temperature, Planet density, Host star temperature, Star luminosity, Star metallicity, Star type

This reduced dimensionality and removed irrelevant or redundant features.

1.4 Column Renaming

All selected features were renamed into readable and meaningful labels to improve interpretability during analysis and modeling.

1.5 Handling Missing Data

Different strategies were applied based on feature type:

Feature Type	Handling Method
Completely empty rows	Removed
Numerical planetary features	Median imputation
Star temperature	Median
Categorical feature (Star type)	Mode

Median imputation was preferred due to robustness against extreme astronomical values.

1.6 Outlier Detection and Physical Validation

Physically impossible values were removed:

- Planet radius ≤ 0
- Planet mass ≤ 0
- Equilibrium temperature ≤ 0 K

Outliers in planet radius were further handled using the Inter-Quartile Range (IQR) method to remove extreme values.
This ensured that only physically meaningful exoplanets were retained.

1.7 Unit Standardization

All features were already provided in standard astronomical units. Therefore, no additional unit conversion was required.

1.8 Feature Engineering

Habitability Score Index
A composite habitability score was created using:

- Proximity to Earth-like temperature (288 K)
- Similarity to Earth-like radius
- Distance from the host star

This transformed astrophysical relationships into machine-learning-interpretable signals.
Orbital Stability Factor
Calculated using:

$$\text{Orbital Stability} = \frac{\text{Orbital Period}}{\text{Semi-major Axis}}$$

Stable orbits contribute positively toward long-term habitability.

1.9 Feature Scaling

- Infinite values introduced during feature engineering were handled safely
- Numerical features were standardized using StandardScaler
- Scaling ensured equal contribution of all variables during model training

1.10 Final Preprocessed Dataset

The cleaned and transformed dataset was saved as: **preprocessed.csv**
This dataset is fully machine-learning ready and used as input for model training.

---

2. Machine Learning Model Training

2.1 Problem Formulation

- Learning Type: Supervised Learning
- Prediction Type: Binary Classification

Label	Meaning
1	Potentially Habitable
0	Non-Habitable

The target variable was created using equilibrium temperature proximity to habitable conditions.

2.2 Dataset Preparation

- Features (X) and target (y) were separated
- Non-predictive identifiers such as planet name and host name were removed
- Dataset was split into:
    - 80% training data
    - 20% testing data
- Stratified sampling ensured class balance

2.3 Preprocessing Pipeline

A unified pipeline was created using scikit-learn Pipelines to prevent data leakage:
- Numerical features → StandardScaler
- Categorical features → One-Hot Encoding
- Model training integrated into the same pipeline

This ensured consistent preprocessing during training and inference.

2.4 Baseline and Primary Models

The following models were trained:
1. Logistic Regression
    - Baseline linear classifier
2. Random Forest Classifier
    - Handles non-linear relationships
    - Robust to noise
3. XGBoost Classifier
    - High-performance gradient boosting model
    - Effective for structured tabular data

2.5 Model Evaluation

Each model was evaluated using multiple metrics: Accuracy, Precision, Recall, F1-Score, ROC-AUC
Additional outputs included:

- Confusion matrix
- Classification report

F1-score was given priority due to class imbalance concerns.

2.6 Hyperparameter Tuning

Hyperparameter tuning was performed for the Random Forest model using GridSearchCV, optimizing:

- Number of trees
- Maximum depth
- Minimum samples split

This improved generalization and reduced overfitting.

2.7 Model Selection

All trained models were compared based on:

- F1-score
- Recall
- Stability on test data

The model with the highest F1-score was selected as the final model.

2.8 Model Saving

The best performing model was saved using joblib as: **best_exohabit_model.pkl**
This allows direct reuse during deployment or inference.

3. Habitability Ranking

Using predicted probabilities from the final model:

- Habitability probability was generated for each exoplanet
- Planets were ranked from most to least habitable

The output file was saved as: **habitability_ranked.csv**

5. Model Interpretability

Feature importance was extracted from the final model:

- Top contributing features were identified
- A feature importance plot was generated
- Scientific interpretation was performed to understand dominant habitability factors

This step improves transparency and trust in model predictions.