Exoplanet Habitability Prediction using Machine Learning

Milestone_2

Name : Ankitha R

Data pre- processing

The data preprocessing stage was one of the most important and challenging parts of this project, as the raw exoplanet dataset was not directly suitable for machine learning. The dataset contained missing values, noisy observations, extreme outliers, and features measured on very different scales. Therefore, a structured preprocessing pipeline was required before moving to model training.The first step involved loading the raw dataset and selecting only the most relevant planetary and stellar features related to habitability. These included planet radius, planet mass, orbital period, semi-major axis, equilibrium temperature, planet density, and key host star properties such as stellar temperature, luminosity, metallicity, and spectral type. Selecting only these features helped reduce noise and focus the model on scientifically meaningful attributes.

After feature selection, completely empty rows were removed from the dataset to avoid unnecessary computation on unusable data. The next major challenge was handling missing values. Since astronomical datasets often have incomplete observations, missing values were handled carefully. For numerical features such as planet radius, mass, orbital period, equilibrium temperature, density, and stellar parameters, median imputation was used. Median was preferred over mean because it is more robust to extreme values. For the categorical feature representing star spectral type, mode imputation was applied. Once missing values were addressed, physical sanity checks were performed. This step was crucial to remove physically impossible values such as negative planet radius, negative mass, or non-positive temperature values. Any rows violating basic physical constraints were filtered out to ensure scientific validity. Outlier detection was then applied using the Z-score method across numerical features. Data points with extreme Z-scores were removed, as they could disproportionately influence the learning process and lead to unstable models. This step helped in stabilizing the dataset while still preserving realistic variations.

Feature engineering was an important part of preprocessing. A habitability index was created by combining planet equilibrium temperature proximity to Earth, similarity of planet radius to Earth, and orbital distance. This feature helped convert astrophysical intuition into a numerical signal that the machine learning model could easily learn from. In addition, a stellar compatibility index was created using host star temperature and luminosity to capture how suitable the star is for supporting potentially habitable planets. An orbital stability factor was also derived from the orbital period, representing the long-term stability of planetary orbits.

The target variable was then created as a binary classification label. Planets falling within predefined ranges of temperature, radius, and orbital distance were labeled as potentially habitable (1), while all others were labeled as non-habitable (0). This transformed the problem into a supervised binary classification task.Since machine learning models are sensitive to feature scales, feature scaling was applied using standardization so that all numerical features had comparable ranges. This ensured that no single feature dominated the learning process due to its magnitude.

Finally, to meet the project constraint, the dataset was restricted to exactly 5000 rows after preprocessing. Random sampling was used to preserve the overall data distribution without introducing bias. The final cleaned and processed dataset was saved as preprocessed.csv, which served as the input for the machine learning model training phase.

Model Training

In this project, machine learning techniques were used to predict the habitability potential of exoplanets based on planetary and stellar features. The implemented code follows a supervised learning approach and treats the problem as a binary classification task.

The preprocessed dataset was first loaded and divided into input features and the target variable. The target variable represents habitability, where a value of 1 indicates a potentially habitable exoplanet and 0 indicates a non-habitable exoplanet. To evaluate the model's ability to generalize to unseen data, the dataset was split into training and testing sets using an 80:20 ratio with a fixed random state.

Initially, baseline machine learning models were trained to establish a reference performance. Logistic Regression was used as a simple linear classifier to understand the basic relationship between features and habitability. A shallow Decision Tree model was also trained to capture non-linear patterns while controlling model complexity. These baseline models helped in comparing the performance of more advanced algorithms.

After establishing baseline results, ensemble-based models were trained to improve prediction accuracy. A Random Forest Classifier was implemented to handle complex feature interactions and reduce overfitting through ensemble learning. Additionally, an XGBoost Classifier was trained due to its strong performance on structured tabular datasets and its ability to model complex relationships between features.

To ensure proper preprocessing and avoid data leakage, feature scaling and model training were combined using scikit-learn Pipelines. This ensured that scaling parameters were learned only from the training data and consistently applied during testing.

The performance of each model was evaluated using multiple metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Confusion matrices and ROC curves were also generated to visualize classification performance. Among these metrics, recall and F1-score were prioritized, as correctly identifying potentially habitable planets is more critical than maximizing overall accuracy. Based on the evaluation results, the Random Forest Classifier was selected as the final model. It achieved an accuracy of approximately 90% and demonstrated stable performance across evaluation metrics. An additional advantage of Random Forest is its ability to provide feature importance scores, which helps in understanding the most influential factors affecting habitability prediction.

Finally, the trained Random Forest model was saved in .pkl format using the joblib library. Saving the model allows it to be reused later without retraining and supports further analysis or deployment.