ExoHabitAI

Milestone 3 - Flask Backend API

Name: Satya Sri Dheeraj M
Date: 01 February 2026

1. Architecture

The backend follows a modular architecture separating concerns between application logic, utility functions, and model inference. The Flask application app.py handles HTTP request routing and response formatting, while utils.py encapsulates input validation, feature engineering, and prediction formatting logic. The trained model is loaded once at startup and cached in memory for efficient inference.

2. Implementation Components

a. Core Files

app.py serves as the main application entry point, defining five primary endpoints: root documentation, health checks, example payloads, single prediction, and ranked retrieval. The application initializes by loading the calibrated Logistic Regression model from `models/final_model_scientific.pkl` and the pre-computed habitability rankings from CSV. Error handling is implemented at both the route level and through Flask's global error handlers to provide meaningful responses for invalid inputs and server failures.

utils.py provides the data processing pipeline required for model inference. Input validation ensures all required fields are present and within physically valid ranges based on astrophysical constraints. The feature preparation function transforms raw input parameters into the 15-feature vector expected by the model, handling one-hot encoding for categorical variables. Response formatting structures predictions into a standardized JSON schema with habitability categories, confidence metrics, and observation recommendations.

generate_ranking.py creates the ranked candidate list by applying the trained model to the full processed dataset. This script removes all features that were excluded during training to prevent data leakage, handles missing values through median imputation, and generates probability scores for all 4,027 unique exoplanets in the database.

b. API Endpoints

The /predict endpoint accepts POST requests containing nine required parameters describing an exoplanet's orbital, physical, and stellar characteristics. After validation, the input is transformed into model features, predictions are generated, and results are returned with habitability probability, binary classification, priority category, and observation recommendations. The endpoint returns 400 for invalid inputs and 500 for prediction failures.

The /rank endpoint retrieves pre-computed habitability rankings via GET requests, supporting optional query parameters for result count and probability thresholds. This endpoint enables efficient retrieval of top candidates without requiring real-time model inference for the entire dataset. Results include rank, planet name, probability, and discovery metadata.

The /batch_predict endpoint processes up to 100 exoplanets in a single request, returning predictions for all valid inputs while reporting errors for invalid entries. This bulk processing capability supports applications requiring analysis of multiple candidates simultaneously.

Supporting endpoints include /health for monitoring model availability, /examples for obtaining valid test payloads, and / for API documentation.

3. Data Flow

Request processing follows a consistent pipeline: HTTP request reception, JSON parsing and content-type validation, input parameter validation against physical constraints, feature vector construction through one-hot encoding and normalization, model inference generating probability scores, response formatting with categorization logic, and JSON serialization for HTTP response.

The ranking system operates differently, loading pre-computed results from disk rather than performing inference. This design choice optimizes performance for queries requiring ordered candidate lists, which represent a common use case for telescope time allocation decisions.

4. Model Integration

The backend interfaces with a calibrated Logistic Regression classifier trained using star system-aware cross-validation to prevent data leakage. The model accepts 15 features: six numerical parameters (orbital period, semi-major axis, planet mass, stellar metallicity, surface gravity, discovery year) and nine binary

categorical indicators for stellar type and planet type. Feature engineering during preprocessing removed all direct and proxy features that were used in creating the binary habitability labels, ensuring the model learns from indirect correlations rather than memorized rules.

Probability calibration using Platt scaling ensures that predicted probabilities reflect genuine uncertainty rather than arbitrary scores. This enables meaningful interpretation of confidence levels and supports decision-making for observation prioritization based on quantified risk.

5. Validation and Error Handling

Input validation enforces both type constraints and physical validity. Numerical parameters must fall within astrophysically reasonable ranges: orbital periods between 0.1 and 100,000 days, semi-major axes from 0.001 to 1,000 AU, planet masses from 0.01 to 13,000 Earth masses, stellar metallicity from -3.0 to 1.0 [Fe/H], surface gravity from 0.0 to 6.0, and discovery years from 1990 to 2030. Categorical parameters accept only predefined values corresponding to spectral types (F, G, K, M, Other) and planet classifications (rocky, super_earth, neptune, jupiter).

Error responses provide actionable feedback distinguishing between client errors (400 for invalid input) and server errors (500 for model failures). Messages include specific parameter names and expected ranges to facilitate debugging and correct usage.

6. Performance Characteristics

The model achieves 87.0% cross-validated recall with conservative estimates due to extreme class imbalance (0.67% habitable planets). On the test set, ranking quality metrics demonstrate practical utility: Recall@10 of 42.9% indicates that 3 of 7 habitable planets appear in the top 10 ranked candidates, while Recall@50 of 85.7% captures 6 of 7 habitable planets within the top 50 recommendations. These metrics reflect the model's effectiveness at prioritizing candidates for follow-up observation despite weak binary classification performance.

Average precision of 0.467 indicates that the probability scores provide meaningful ranking information even when absolute classification accuracy is limited by data constraints and class imbalance.

7. Deployment Configuration

The development server runs on host 0.0.0.0 port 5000 with debug mode enabled for detailed error reporting.

Environment requirements include Flask 3.0, scikit-learn 1.4, XGBoost 2.0, pandas 2.0, and numpy 1.24. These dependencies are specified in the project requirements file for reproducible deployment.

Scientific Considerations

The API returns predictions with appropriate caveats embedded in response categorization. Low-probability planets are marked as "Not Habitable" with recommendations against follow-up observation, while high-probability candidates receive "High Priority" designation with suggestions for immediate spectroscopic analysis. This categorization helps non-expert users interpret raw probability scores in the context of observational decision-making.

Confidence levels are derived from the distance between predicted probability and the decision boundary, providing a heuristic measure of prediction certainty. Models with probabilities near 0.5 receive "Low" confidence designations, while extreme probabilities yield "High" confidence markers.

The ranking system acknowledges that habitability labels are based on simplified habitable zone criteria rather than confirmed observations of liquid water or biological activity.

Conclusion

This backend implementation provides a robust, scientifically grounded interface to the exoplanet habitability prediction model. The API design balances ease of use through comprehensive documentation and examples with rigorous validation to prevent misuse. The architecture supports both single-planet analysis for newly discovered candidates and bulk retrieval of ranked priorities for systematic observation planning, addressing the primary use cases in astronomical research workflows.