Exoplanet Candidate Scoring and Target-Level Diagnostics: A Mathematical, Astrophysical, and Computational Description of `exo_analysis.py`

1 Goal and Setting

The program `exo_analysis.py` ingests three NASA Exoplanet Archive–style tables:

- TESS Objects of Interest (TOI),
- Kepler Objects of Interest (KOI),
- K2 planet candidates.

It trains a supervised/PU (positive–unlabeled) ensemble on any dataset that has labeled dispositions, produces calibrated probabilities $p \in [0, 1]$ that a target is a real exoplanet, and generates per-target diagnostic figures (feature standardization, local contributions, Monte–Carlo uncertainty, PCA neighborhood). It also exports dataset-level diagnostics (ROC/PR, reliability, PCA overview) and CSVs with predictions and reasoning.

Mathematically, the method maps raw astrophysical attributes $\mathbf{x} \in \mathbb{R}^d$ to a calibrated score

$$p = \Pr(\text{planet} \mid \mathbf{x}),$$

together with uncertainty and explanation terms.

2 Astrophysical Feature Model

Let $R_\star$ be the stellar radius (in $R_\odot$), T_{eff} the stellar effective temperature (K), P the orbital period (days), d the transit depth (ppm), D the transit duration (hr), b impact parameter, and a the semi-major axis.

Core derived features. The script builds dimensionless and scale-stable features:

$$\text{Depth fraction:} \quad f_d \;=\; \frac{d}{10^6} \tag{1}$$

$$\text{Radius ratio proxy:} \quad \frac{R_p}{R_\star} \;\approx\; \sqrt{f_d} \tag{2}$$

$$\text{Scaled semi-major axis:} \quad \frac{a}{R_\star} \;\approx\; \left[\frac{GM_\star}{(2\pi/P)^2}\right]^{1/3} \frac{1}{R_\star} \;\propto\; \frac{P^{2/3}}{R_\star} M_\star^{1/3} \tag{3}$$

$$\text{Duration aspect:} \quad \alpha_R \;=\; \frac{D}{P} \quad \text{and} \quad \frac{D}{P^{1/3}} \quad \text{(heuristics for transit geometry/SNR)} \tag{4}$$

$$\text{Insolation proxy:} \quad S \;\propto\; \frac{R_\star^2 T_{\text{eff}}^4}{a^2} \tag{5}$$

$$\text{Equilibrium-}T\text{ proxy:} \quad T_{\text{eq}} \;\approx\; \frac{T_{\text{eff}}}{\sqrt{2}} \sqrt{\frac{R_\star}{a}} \tag{6}$$

When stellar mass $M_\star$ is unknown, the code uses a weak power-law $M_\star \propto R_\star^\gamma$ (typical $\gamma \in [0.8, 1.2]$) or omits the exact dependency and treats $\frac{a}{R_\star}$ as a robust period-radius scaling feature.

Quality and magnitude features. SNR proxies (model_SNR, d/σ_d), band magnitudes (e.g. `Vmag`, `Kmag`, `G`), and flags (when available) are included as independent coordinates.

Robust standardization. For each feature x_j the standardized value is

$$z_j \;=\; \frac{x_j - \text{med}(x_j)}{\text{IQR}(x_j) + \epsilon},$$

with $\epsilon > 0$ to avoid division by zero. Median/IQR are robust to outliers and missingness patterns.

3 Labels and Learning Regimes

Let $y \in \{0, 1, \varnothing\}$ denote {false positive, confirmed/candidate, unlabeled}.

- **Supervised** (e.g. TOI): many $y \in \{0, 1\}$; unlabeled rows are ignored for fitting, used for scoring.
- **Positive–Only** (e.g. KOI when no FPs): the script falls back to a one-class style. Concretely, it:
  1. Fits a *stack* on positives vs. a large internal synthetic background (§4).
  2. Calibrates scores to $\hat{p}$ with isotonic or Platt scaling using cross-validation within positives (treating folds as pseudo-negatives) and reliability constraints.
- **PU Bagging** (optional when some unlabeled exist): repeatedly sample pseudo-negatives from unlabeled, fit a base classifier, and average predicted posteriors:

$$\hat{p}_{\text{PU}}(\mathbf{x}) \;=\; \frac{1}{B} \sum_{b=1}^{B} f_b(\mathbf{x}), \tag{7}$$

  with $B = O(10-50)$ depending on dataset size.

4 Stacked Classifier and Calibration

The stack comprises diverse base learners on standardized features $\mathbf{z}$:

$$\{f^{(m)}(\mathbf{z})\}_{m=1}^{M}$$

where m indexes models such as:

- Logistic Regression (LR) with class weights,
- Gradient Boosting (GB) / Histogram Gradient Boosting,
- Random Forest (RF),
- Support Vector Machine (SVM, RBF kernel; posterior via Platt scaling).

The level-2 combiner is a calibrated LR (or GB) on the out-of-fold base predictions:

$$s(\mathbf{z}) \;=\; \sum_{m=1}^{M} w_m \, f^{(m)}(\mathbf{z}) + b, \qquad \hat{p} \;=\; \sigma\!\left(\frac{s-\mu}{\sigma_s}\right)$$

with μ, σ_s from a held-out calibration set. The final probability uses either:

$$\text{Platt (sigmoid):} \quad \hat{p} \;=\; \sigma(as + c), \tag{8}$$

$$\text{Isotonic:} \quad \hat{p} \;=\; \mathrm{Iso}(s), \tag{9}$$

choosing the variant with smaller Brier score on validation.

Reliability. Given K bins $\{B_k\}$ over $[0, 1]$, the expected calibration error (ECE) is

$$\mathrm{ECE} \;=\; \sum_{k=1}^{K} \frac{|B_k|}{N} \left| \mathrm{acc}(B_k) - \mathrm{conf}(B_k) \right|.$$

Reliability plots and ECE are reported per dataset.

5 Uncertainty and Target Diagnostics

Predictive uncertainty. For a target with standardized vector $\mathbf{z}_i$:

1. **Entropy:** $H_i = -\hat{p}_i \log \hat{p}_i - (1 - \hat{p}_i) \log(1 - \hat{p}_i)$.
2. **Margin:** $m_i = \hat{p}_i - 0.5$.
3. **Monte–Carlo perturbation.** Perturb standardized features with independent noise $\eta_j \sim \mathcal{N}(0, \sigma_j^2)$ where $\sigma_j = \lambda \cdot \mathrm{IQR}(z_j)$ with small λ (e.g. 0.05). The Monte–Carlo posterior $\{\hat{p}_i^{(r)}\}_{r=1}^{R}$ yields mean, st. dev., and empirical quantiles $(5\%, 95\%)$.

Local contribution (linear surrogate). Around $\mathbf{z}_i$, fit a ridge surrogate $\tilde{s}(\mathbf{z}) = \boldsymbol{\beta}_i^{\top} \mathbf{z} + c$ on k nearest neighbors in PCA space. The top-q contributors are the largest $|\beta_{i,j} z_{i,j}|$.

Counterfactual proxy. Solve a small ℓ_2-regularized least-squares in $\Delta\mathbf{z}$ to reach a target probability τ using the surrogate:

$$\min_{\Delta\mathbf{z}} \|\Delta\mathbf{z}\|_2^2 \quad \text{s.t.} \quad \sigma\big(\boldsymbol{\beta}_i^\top(\mathbf{z}_i + \Delta\mathbf{z}) + c\big) \geq \tau,$$

giving interpretable "how much to change" suggestions in standardized units.

6 Evaluation Metrics

For labeled sets, the script computes:

$$\begin{aligned}
\text{ROC AUC:} \quad \mathcal{A}_{\text{ROC}} &= \int_0^1 \text{TPR}(\text{FPR})\, d\,\text{FPR}, && (10)\\
\text{PR AUC:} \quad \mathcal{A}_{\text{PR}} &= \int_0^1 \text{Prec}(\text{Rec})\, d\,\text{Rec}, && (11)\\
\text{Brier:} \quad \mathcal{B} &= \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - y_i)^2, && (12)\\
\text{Youden } J \text{ for threshold } t: \quad J(t) &= \text{TPR}(t) + \text{TNR}(t) - 1. && (13)
\end{aligned}$$

Threshold tables include t maximizing F_1, J, and balanced accuracy.

7 Handling Missing Data

Let x_{ij} be missing. The program applies:

1. Median imputation per feature: $\tilde{x}_{ij} \leftarrow \text{med}(x_j)$.
2. Missingness indicators $m_j = \mathbb{I}\{x_{ij} \text{ missing}\}$ can be included as auxiliary features.
3. Robust standardization uses med/IQR, which tolerates mild missingness.

Base models that accept NaNs natively (e.g. Histogram GB) are prioritized inside the stack for stability.

8 Computational Notes

Scaling. Let n be rows, d features, M base models, B PU bags. Major costs:

Histogram GB: $O(nd\log n)$, RF: $O(M_{\text{trees}}nd)$, SVM (RBF): $O(n^2d)$ (mitigated by subsampling for large n).

The script parallelizes cross-validation folds per model and caches standardization statistics. Target-level MC uses $R \ll 10^3$ draws.

9 Outputs and Directories

For each dataset `TOI/KOI/K2` under the chosen `--out`:

- `dataset_predictions.csv`: {designation, prob, label, entropy, margin, MC stats}.

- `top_candidates.csv`, `threshold_table.csv`, `perf.csv`.
- `fig/` : ROC, PR, reliability, PCA overview.
- `targets/`*`DESIG`*`/`: per-target PNG with standardized features, local contributions, MC histogram, PCA neighborhood, and a JSON (`analysis.json`) containing the machine-readable reasoning (top features, counterfactual deltas).
- `index.html`: links to all reports, with minimal dashboard tiles.

10 Scientific Interpretation Checklist

1. **Transit physics sanity:** high-probability targets should show consistent (P, D, f_d) with feasible $a/R_\star$, $R_p/R_\star$ and $T_{\rm eq}$ relative to $T_{\rm eff}$.
2. **Calibration:** ECE $<$ few% on labeled data; reliability curve close to diagonal.
3. **Class balance**: avoid leakage by stratified CV; use class weights or PU bagging when negatives are sparse.
4. **Astrophysical priors**: the feature set encodes simple transit geometry and radiative scaling; extreme or inconsistent combinations are often down-weighted by the ensemble.

11 Minimal End-to-End Algorithm (Pseudo-code)

$$
\begin{aligned}
&\textbf{for } D \in \{\mathrm{TOI}, \mathrm{KOI}, \mathrm{K2}\}: \\
&\quad T \leftarrow \text{read\_csv}(D) \text{ with header cleanup and comment stripping} \\
&\quad X \leftarrow \text{build\_features}(T); \quad (Z, \mathrm{med}, \mathrm{IQR}) \leftarrow \text{robust\_z}(X) \\
&\quad y \leftarrow \text{labels}(T) \\
&\quad \textbf{if } (\exists\, y \in \{0,1\}): \quad (f_{\text{stack}},\ \text{cal}) \leftarrow \text{train\_stack}(Z, y) \\
&\quad \textbf{else}: \quad f_{\text{stack}} \leftarrow \text{one\_class\_surrogate}(Z) \\
&\quad \hat{p} \leftarrow \text{cal}(f_{\text{stack}}(Z));\ \text{compute ROC/PR/Brier/ECE on labeled} \\
&\quad \textbf{for target } i: \ \text{make local surrogate, MC uncertainty, counterfactuals} \\
&\quad \text{export CSVs + figures + dashboard.}
\end{aligned}
$$

12 Limitations and Extensions

- The derived features use first-order transit and radiative scalings; full light-curve model fits (e.g. Mandel–Agol) are out of scope but could be integrated to refine $R_p/R_\star$, b, and SNR.
- KOI may be overwhelmingly positive; the positive-only calibration provides a *ranking* calibrated to internal reliability but cannot estimate absolute contamination without external negatives or priors.
- Future extension: semi-supervised consistency regularization, astrophysical priors via Bayesian calibration, and joint multi-survey domain adaptation.

13 Reproducibility

Command-line interface:

```
python exo_analysis.py --toi <toi.csv> --koi <koi.csv> --k2 <k2.csv> --out <out_dir>
# optional:
#   --limit-targets N     # analyze first N targets per dataset
#   --seed 7              # reproducible splits/MC
#   --quiet               # terse logging
```