Exo–CAL: Physics–aware, probability–calibrated exoplanet candidate analysis

A readable description of the mathematics, astrophysics, and computing behind the `exo_analysis` program

Exo–CAL Team

Executive summary

What Exo–CAL does. Exo–CAL is a software program that reads open exoplanet candidate tables from well known surveys and produces, for every target, an estimated probability that the signal is caused by a planet. It couples simple and transparent physics with modern, calibrated machine learning. The program also makes human readable figures and comma separated value files so that a scientist and a non specialist can both inspect and share results.

Why this matters. New surveys discover thousands of candidate signals. Human teams cannot manually vet everything. It is valuable to have a tool that can: (i) clean the tables, (ii) compute physically meaningful features, (iii) learn from historical labels where they exist, (iv) produce well calibrated probabilities rather than raw scores, (v) quantify uncertainty, and (vi) explain *why* a probability is high or low.

Inputs and outputs. Inputs are the three common candidate tables (the TESS Object of Interest table, the Kepler Object of Interest table, and the K2 candidate table) as downloaded from the public archive. Outputs are: per target probability files with explanations, small images for each target, whole dataset plots such as precision–recall and reliability, and a light web page that can be opened locally to browse the results.

1 High level flow

The program `exo_analysis.py` follows the pipeline below:

1. **Read and standardise tables.** Lines beginning with the archive comment marker ignored. Column names are normalised while preserving the original names in the metadata so the files remain traceable.
2. **Clean missing and non numeric values.** Units are detected and converted into consistent systems where necessary; numbers embedded as strings are parsed; non finite values are handled by robust statistics.
3. **Create physics–aware features.** For every target we compute quantities that have direct meaning in the transit method: depth as planet radius divided by star radius squared, transit duration scaled by orbital period, orbital separation divided by star radius using Kepler's third law, equilibrium temperature, and others. The definitions appear in Section 2.
4. **Standardise features.** Each feature is transformed into a zero–mean, unit scale variable using a robust centre and a robust spread[1]. This makes learning stable even when a survey has long tails or outliers.

[1]Robust centre is the median, robust spread is the interquartile range divided by one point three four nine.

5. **Learn a predictive model.** A small group of complementary models is trained and stacked: gradient tree boosting, random forest, and a distance based learner. Their scores are combined by a simple linear model learned on a holdout fold.

6. **Calibrate scores into probabilities.** The stacked score is transformed into a true probability using logistic calibration on a held out fold. This step is essential: a score of zero point eight should *mean* that eight of ten similar cases are real.

7. **Quantify uncertainty by Monte Carlo.** For each target we draw random versions of its features using the estimated measurement spread, recompute the probability hundreds of times, and report the mean, standard deviation, and central interval.

8. **Explain the prediction.** We compute a simple, local contribution vector by multiplying standardised features by the learned linear weights. The largest absolute entries form the top contributor list, which is easy to read without specialist language.

9. **Export results and figures.** For every dataset and every target we write a small `summary.json`, a one page image showing the features, the local contributions, the Monte Carlo histogram, and a neighbour plot. Dataset level figures include precision–recall, reliability, and a distribution of predicted probabilities.

2 Astrophysical ingredients

The program uses the simplest physically grounded relations that connect the observed transit signal with planet and star properties. All symbols are defined in place; the reader does not need prior notation.

Transit depth and size ratio. In the planetary transit method the fractional drop in brightness, which we call *depth*, is approximately

$$\text{depth} \approx \left(\frac{R_{\text{planet}}}{R_{\text{star}}} \right)^2 . \tag{1}$$

We therefore compute the ratio $R_{\text{planet}}/R_{\text{star}}$ whenever both radius estimates are present, and we include both depth and the ratio as features because pipelines may report one or the other.

Orbital separation through the period. Kepler's third law relates orbital period P and semi major axis a:

$$a \approx \left(\frac{G M_{\text{star}}}{4\pi^2} \right)^{1/3} P^{2/3}. \tag{2}$$

When star mass is missing we fall back to a photometric mass estimate from star effective temperature and gravity if available. We then form a dimensionless feature

$$\frac{a}{R_{\text{star}}} \quad \text{(orbital separation in units of star radius)}. \tag{3}$$

Transit duration scaled by period. A central transit in a circular orbit has an idealised duration that scales as

$$T_{\text{dur}} \propto \frac{P}{\pi} \arcsin\left(\frac{R_{\text{star}}}{a} \right). \tag{4}$$

The ratio T_{dur}/P and the pair $(T_{\text{dur}}, a/R_{\text{star}})$ provide a simple physical sanity check: very long duration at very small separation is suspicious, very short duration at very large separation is also suspicious.

Equilibrium temperature. A rough estimate of planet equilibrium temperature (no greenhouse and full redistribution) is

$$T_{\mathrm{eq}} \approx T_{\mathrm{eff,\ star}} \sqrt{\frac{R_{\mathrm{star}}}{2a}}\ (1-A)^{1/4}\,, \tag{5}$$

where A is the Bond albedo; we use a neutral fixed value when it is not known. This is not used to judge habitability; it only helps the model understand whether the incident flux is extremely high or extremely low, which often correlates with astrophysical false positives.

Signal quality hints. Reported signal to noise of the transit search, photometric scatter level, and the number of detected transits are harmless and useful context features. The model never requires them, but when present they help separate robust candidates from borderline detections.

3 Mathematical treatment

3.1 Robust standardisation

For each raw feature x the program computes a robust centre m and robust spread s

$$m = \mathrm{median}(x), \qquad s = \frac{\mathrm{quantile}_{0.75}(x) - \mathrm{quantile}_{0.25}(x)}{1.349}.$$

The standardised feature is $z = (x - m)/\max(s, \epsilon)$ with a very small ϵ to avoid division by zero. Robust statistics are preferred because survey tables often have genuine outliers.

3.2 Stacked learning with calibration

Let $Z \in \mathbb{R}^{n \times d}$ be the matrix of standardised features and let $y \in \{0,1\}^n$ be the training labels where available. We fit three base learners that make complementary assumptions:

1. Gradient boosted decision trees: piecewise constant, good for non linear effects.
2. Random forest: bagged trees, strong on interaction and monotonic trends.
3. Distance based learner: probability estimated from neighbours in the feature space.

Each base learner produces a score $s_k(Z) \in [0,1]$. A linear stacker learns weights $w \geq 0$ on a holdout fold and forms

$$s_{\mathrm{stack}}(Z) = \sum_k w_k\, s_k(Z).$$

Scores are then turned into honest probabilities using logistic calibration on the holdout fold:

$$p(Z) = \sigma(a\, s_{\mathrm{stack}}(Z) + b)\,, \qquad \sigma(t) = \frac{1}{1+\exp(-t)}.$$

Calibration is evaluated through a reliability curve: we bin the predicted probabilities and compare the fraction of true positives inside each bin with the average prediction inside the same bin. A nearly diagonal curve indicates good calibration.

3.3 Uncertainty by Monte Carlo

For each target i we approximate feature measurement spread by a diagonal covariance Σ_i. We draw M synthetic versions of the target,

$$Z_i^{(m)} \sim \mathcal{N}(Z_i,\, \Sigma_i), \quad m = 1, \ldots, M,$$

and compute probabilities $p^{(m)} = p\Big(Z_i^{(m)}\Big)$. The program reports

$$\overline{p} = \frac{1}{M} \sum_m p^{(m)}, \quad \text{standard deviation}, \quad \text{fifth percentile}, \quad \text{ninety fifth percentile}.$$

The Monte Carlo histogram appears on each target page so the user sees whether the estimate is sharp or wide.

3.4 Local explanation

The final linear layer in the stack provides a very simple explanation proxy: for a target with standardised features z and linear weights w, we compute the vector

$$c = z \odot w,$$

where $\odot$ denotes elementwise multiplication. The largest magnitudes indicate which standardised features are pushing the probability up or down. This is not a full game theoretic explanation, but it is stable and easy to read.

4 Computing and data handling

4.1 Input tables

The program accepts the three well known public tables as comma separated files. The parser:

- skips the archive header lines beginning with a hash sign,
- preserves original column names in the metadata,
- supports both numeric columns and numeric strings,
- tolerates missing values without dropping rows.

4.2 Feature engineering in practice

A survey may not report every physical quantity. The program chooses the best available source following a hierarchy. For radius ratio it prefers an explicit ratio; if absent it builds it from planet radius and star radius, and if only transit depth is available it takes the square root of depth. The same idea applies to orbital separation: if star mass is missing, a photometric proxy is used; if it is absent, the feature is marked as not available and the model handles it.

4.3 Evaluation for whole datasets

When historical labels exist the program reports:

- precision–recall curve with area under the curve,
- true positive rate versus false positive rate curve with area under the curve,
- reliability (also called calibration) curve with the Brier score.

When only positive labels exist (for example some versions of the Kepler Object of Interest table) the program switches to a conservative positive–unlabelled setting: the negative class is treated as unknown and only ranking, not absolute accuracy, is reported. The program prints a clear message when this mode is used.

5 Outputs and how to read them

Per dataset folder

For each of the three datasets Exo–CAL writes a folder that contains:

- `dataset_predictions.csv`: one line per target with designation, calibrated probability, entropy of the prediction.
- `top_candidates.csv`: the highest ranked subset for quick review.
- `perf.json` and small images: precision–recall curve, true positive rate curve, reliability curve, and a two dimensional principal component projection used for orientation only.
- `threshold_table.csv`: operating points that trade off completeness and purity. Each line gives the decision threshold together with expected precision and expected recall estimated by cross validation.

Per target folder

Inside `targets/<designation>` the program writes:

- `summary.json`: designation, dataset name, calibrated probability, entropy, Monte Carlo summary, a list of the five strongest contributors, and key physical values (period, duration, depth in parts per million, planet radius in Earth radius units, star effective temperature, star radius).
- `panel.png`: a one page figure with four panels: standardised features, local contributions, Monte Carlo probability histogram, and a principal component scatter with the target marked. The scatter gives a qualitative sense of whether the target lives in a region with many previous positives.

6 Design choices and justification

Physics first. The best predictor for transit detections is still the geometry and timing of the transit itself. Features that reflect simple relations are strong and understandable, and they remain valid across instruments.

Probability calibration. Many predictive tools emit a score whose scale is arbitrary. Exo–CAL insists on a final step that maps scores to probabilities so a user can combine them with costs and benefits. Probability is the quantity scientists and decision makers can actually use.

Transparency and stability. The program prefers small numbers of robust features, simple linear stacking, and well known classifiers. Explanations are intentionally simple. This stability helps the user trust the result.

Conservative use of labels. When only positive labels exist, declaring the rest negative would be misleading. In that case Exo–CAL trains a one sided model that ranks by similarity to positives and prints a visible message explaining that absolute accuracy cannot be computed.

7 How to interpret a single target page

A practitioner can skim a target panel in less than a minute:

1. **Calibrated probability and entropy.** A probability near one is confident. Entropy close to zero means very confident; entropy near one bit means very uncertain.
2. **Top contributors.** The short list tells what pushed the estimate. For example a physically reasonable duration to period ratio at a moderate size ratio often pushes up; an extremely small duration at a very large separation often pushes down.
3. **Monte Carlo histogram.** A narrow histogram shows that small changes in the measured values do not change the conclusion; a wide histogram signals that the decision is sensitive to measurement uncertainty.
4. **Neighbour projection.** The principal component scatter is only a rough orientation tool, not a proof. If the point sits inside a cloud of previous positives that increases confidence; if it is an isolated outlier that decreases confidence.

Appendix: list of features

For transparency, the feature vector includes the following when present in the source tables:

- orbital period, transit duration, transit depth in parts per million,
- planet radius, star radius, star effective temperature,
- ratio of planet radius to star radius, square root of depth,
- orbital separation divided by star radius, estimated from period and star mass,
- duration divided by period,
- equilibrium temperature,
- logarithm of separation ratio and logarithm of star gravity when present,
- signal to noise as reported by the pipeline.