**Final\_models\_k2.py**

**Step-by-Step Explanation of the Code**

**1. Data Loading and Preparation**

* The code loads a CSV file containing k2 data
* It creates a new column target to encode the planet's status: 1 for CONFIRMED, 0 for FALSE POSITIVE/REFUTED, and -1 for CANDIDATE.
* The data is split into two groups: labeled (known exoplanet or not) and candidates (unknown status).

**2. Feature Selection and Preprocessing**

* Columns that are not useful for machine learning (like names, IDs, and references) are excluded from the feature list.
* Missing values in features are filled with the median value from the labeled data.
* Features are scaled (standardized) so all have similar ranges.

**3. Train/Test Split**

* The labeled data is split into training and test sets, keeping the class balance similar in both (stratified split).

**4. Model Training**

* **Random Forest Classifier**: The training data is balanced using SMOTE (a technique to create synthetic examples of the minority class), then a Random Forest model is trained on this balanced data.
* **Neural Network (MLPClassifier)**: A neural network is trained on the original (unbalanced) training data.

**5. Model Evaluation**

* Both models are evaluated on the test set using F1 score (how well the model balances precision and recall) and ROC AUC (how well the model separates the two classes at all thresholds).

**6. Candidate Prediction and Habitability Check**

* The Random Forest model predicts which candidate planets are likely exoplanets.

For those predicted as exoplanets, the code checks if they are potentially habitable by scoring them on:

**1. Radius ()**

* **Check:**  Earth radii
* **Why:** Planets in this range are likely to be rocky (like Earth) or water-rich, not gas giants. Too small, and they may lose their atmosphere; too large, and they are likely gaseous mini-Neptunes.

**2. Mass ()**

* **Check:**  Earth masses
* **Why:** Similar to radius, this range favors planets that can hold an atmosphere and have a solid or water-rich surface, but are not gas giants.

**3. Equilibrium Temperature ()**

* **Check:**  Kelvin
* **Why:** This is the temperature range where liquid water could exist on the surface, which is essential for life as we know it.

**4. Insolation Flux ()**

* **Check:**  (relative to Earth)
* **Why:** This is the amount of energy the planet receives from its star. Too little, and water freezes; too much, and water boils away. This range covers the "habitable zone".

**5. Host Star Effective Temperature ()**

* **Check:**  Kelvin
* **Why:** This range includes M-dwarfs to Sun-like stars, which are stable enough for life to develop and have well-defined habitable zones.

**6. Eccentricity ()**

* **Check:**  (or missing)
* **Why:** Low eccentricity means a more stable climate, reducing extreme temperature swings that could make life difficult

Each candidate gets a habitability score (fraction of criteria met). The code prints the top candidates with the highest scores.

**7. Visualization**

* The code plots ROC curves for both models, showing how well they distinguish exoplanets from non-exoplanets at different thresholds.

**habitability\_for\_all.py**

Habitability Check for all CONFIRMED exoplanets in k2 dataset.