Orbitify

Executive Summary:

Orbitify aims to solve the challenge of reliably and accurately identifying new exoplanets, given the challenges of noisy telescope signals, overlapping stellar activity, and the vast amount of observational data to analyze. Our solution is to build a Machine Learning pipeline where we automate the process of signal preprocessing, feature engineering and extraction, and classification whether the data is a candidate exoplanet or not. We also provide a user interface to be able to interact and explore the data.

Problem Definition:

Exoplanet Detection expands our understanding of planetary systems, uniqueness of our planetary system and identifying habitable planets, current methods face significant obstacles. Data from space missions such as Kepler is massive, noisy, and often contaminated by stellar variability, making reliable identification difficult. Inaccuracies or inefficiencies in the detection pipeline can delay discoveries, misallocate research resources, and hinder scientific progress, that's why a solution capable of automating this whole pipeline reliably is crucial for the advancements in this field.

Background & Literature Review:

Research in exoplanet detection has evolved from manual vetting to sophisticated machine learning methods. Early approaches applied ensemble algorithms such as Random Forests and Gradient Boosting on features engineered from Kepler and TESS light curves, demonstrating improved accuracy over manual or rule-based classifiers. reference

Deep learning significantly advanced the field. Convolutional Neural Networks (CNNs), most notably in Shallue & Vanderburg's Astronet, directly learned from raw or minimally processed light curves, successfully recovering previously overlooked planets. reference

Recent work has explored biologically inspired models. ExoSpikeNet (Chatterjee et al., 2024) introduced Spiking Neural Networks (SNNs) for Kepler light curves, leveraging temporal dynamics for enhanced detection. The model achieved strong performance achieving 99% accuracy, suggesting that neural architectures inspired by brain activity can outperform conventional approaches for time-series signals. Reference

We propose leveraging raw light curve data for supervised learning. Instead of relying solely on preprocessed datasets, we extract statistical and informative features directly from the light curves. By aligning these features with confirmed classifications, we aim to train a machine learning model capable of discriminating true planetary transits from stellar variability and instrumental noise, ultimately advancing automated exoplanet detection beyond current baselines.

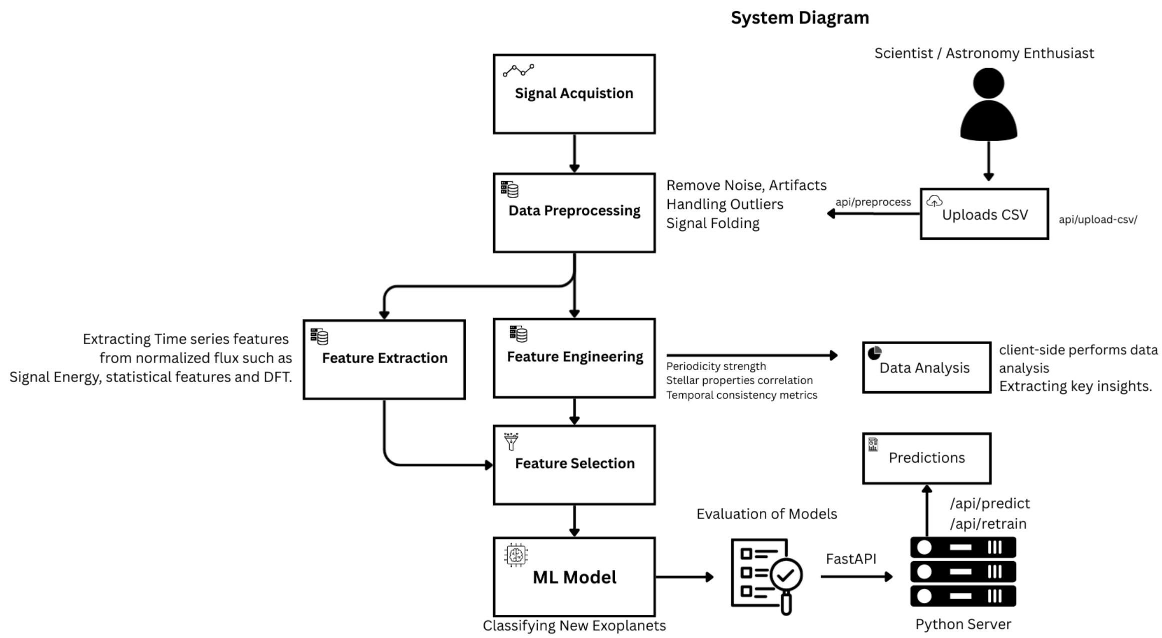
Methodology:

Our methodology is designed to develop an automated pipeline for exoplanet detection that combines data preprocessing, machine learning, and a user-friendly web interface. The process is divided into sequential steps

- Gather light curve datasets from https://archive.stsci.edu/kepler/publiclightcurves.html
- Retrieve confirmed planet candidates and false positives from the Exoplanet Archive to serve as labeled data for supervised learning.
- Remove systematic noise
- Apply data preprocessing to prepare for feature extraction
- Extract statistical features such as mean flux, variance, skewness, and periodicity indicators.
- Derive astrophysical transit-specific features, including transit depth, duration, and shape.
- Use Fourier and wavelet transforms to capture frequency-domain characteristics and periodic stellar variability.
- Establish baselines with classical models (Random Forest, Gradient Boosting) trained on extracted features.
- Employ deep learning architectures to directly learn patterns from raw or minimally processed light curves.
- Compare traditional feature-based models with end-to-end deep learning models to identify the most effective detection pipeline.
- Apply model evaluation and validation
- Refine and hyperparameter tune
- Benchmark results against existing models
- Wrap the trained model in a FastAPI service that can receive new light curve data and return classification results.
- Build an interactive frontend (Next.js) to visualize light curves, highlight detected transits, and allow users to explore results.
- Document the entire pipeline from preprocessing to model selection.

Solution:

Orbitify is an end-to-end machine learning pipeline designed to automate exoplanet detection from raw light curve data. The system integrates data preprocessing, feature extraction, model training, and a web-based interface to provide both researchers and astronomy enthusiasts with a reliable tool for analyzing exoplanet candidates.



Features:

1. Automated Signal Preprocessing
2. Visualization and extracting key insights
3. Interface for uploading raw signals.
4. Batch Predicting
5. Model Retraining on your own data.

Value Proposition:

Orbitify reduces the time and effort researchers spend on cleaning and analyzing noisy light curves by providing an automated, reproducible detection pipeline. It minimizes false positives and accelerates the discovery process, enabling scientists to focus on interpreting results. For astronomy enthusiasts and students, the interactive interface makes real exoplanet data accessible and engaging, turning complex signals into clear visualizations.

Role of Team Members:

Yousef Mohamed Farouk	Designing Web Interface. (Primary)	Working on the Machine Learning Model. (Secondary)
Anas Taha Yahya	Working on the Machine Learning Model. (Primary)	Video Editing (Secondary)
Mohamed Ahmed Mostafa	Working on the data preprocessing Pipeline. (Primary)	

Workflow Strategy:

Setup:

- Create GitHub repo and set collaboration rules.
- Install all required packages and tools.
- Build the basic structure of the Next.js web interface.
- Collect and prepare the dataset.
- Do initial preprocessing, visualizations, and start feature extraction.
- **Milestone 1 Deliverables:**
    - Web interface skeleton.
    - Clean dataset with some visual insights.
    - Early feature extraction scripts.
- Train models, compare results, and refine features.
- Evaluate performance and tune hyperparameters.
- Deploy the model with FastAPI and connect it to the backend.
- Finalize the web interface and show results.
- Write short documentation of the process and findings.
- **Milestone 2 Deliverables:**
    - Trained and evaluated model.
    - Hosted API endpoints.
    - Integrated web application.
    - Final documentation.

Resources:

List all the resources, including software, hardware, data, and references, that have been utilized in your project. Make sure to include at least one scientific paper in your references.

Resources:

- https://science.nasa.gov/exoplanets/how-we-find-and-characterize/
- https://home.strw.leidenuniv.nl/~keller/Teaching/ADA_2011/ADA_2011_L06_Exoplanets.pdf
- https://heasarc.gsfc.nasa.gov/docs/tess/
- https://www.planetary.org/articles/color-shifting-stars-the-radial-velocity-method
- https://exoplanetarchive.ipac.caltech.edu/docs/rv.html\

Data:

- https://archive.stsci.edu/kepler/publiclightcurves.html
- https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative
- https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=TOI
- https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=k2pandc
- https://www.kaggle.com/datasets/vijayveersingh/kepler-and-tess-exoplanet-data?select=TOI_2025.02.03_06.18.31.csv

Research:

- https://www.mdpi.com/2079-9292/13/19/3950
- https://academic.oup.com/mnras/article/513/4/5505/6472249

Write the team name and members names.

Team Name :

1- Yousef Mohamed Farouk

2- Anas Taha Yahya

3- Mohamed Ahmed Mostafa