**Project Report: AI-Based Exoplanet Detection**

**Project Title:** Planetary Detection AI — Classifying Exoplanets Around Stars Using Machine Learning

**Author:** Som Subhro Nath, B.Sc. (CS with Data Science), University of Calcutta.  
**Submission:** NASA Space Apps Challenge 2025

**1. Introduction**

The discovery of exoplanets (planets outside our solar system) has traditionally relied on manual analysis of astronomical data collected by space telescopes. With the exponential growth in data from missions such as Kepler and TESS, manual inspection is no longer feasible.

This project leverages **Machine Learning (ML)** to automate exoplanet detection, providing probabilistic classification of stars into planet-hosting or non-planet-hosting categories. The system integrates **XGBoost** for modelling and **FastAPI** for deployment, ensuring accessibility and scalability.

**2. Problem Statement**

* Millions of stars are observed, but only a fraction are analyzed in detail.
* Traditional methods are labour-intensive, slow, and prone to bias.
* Large datasets (>TBs) cannot be processed efficiently on conventional PCs.

**Goal:**  
To build an AI-based system capable of automatically predicting whether a given star hosts planets, while being reproducible, non-biased, and scalable.

**3. Data Collection**

* **Source:** Public datasets available via Kaggle and NASA repositories.
* **Content:** Stellar parameters such as kepid, ra, dec, kepmag, teff, logg, feh, radius, mass, and density.
* **Preprocessing:**
  + Cleaning missing or corrupt entries.
  + Normalization and noise reduction.
  + Merging multiple datasets to create a unified training set (~4GB).
  + SMOTE applied to reduce class imbalance.

Note: Full datasets exceed multiple TBs; only a representative subset was used for feasibility on standard hardware.

**4. Methodology**

**4.1 Feature Engineering**

* Numerical features (e.g., teff, radius) were scaled.
* Categorical features were encoded as needed.
* Target column: nconfp — binary label: 0 (no planet), 1 (planet present).

**4.2 Model Selection**

* **XGBoost Classifier:** Chosen for its:
  + Robustness to imbalanced datasets.
  + Superior performance on tabular data compared to deep learning models for moderate-sized datasets.
  + Efficiency and interpretability.
* Deep learning (PyTorch) was considered but discarded due to:
  + Tabular nature of data (XGBoost outperforms neural networks in tabular tasks).
  + Limited computational resources.
* LightGBM was considered but XGBoost offered better handling of outliers and more mature ecosystem.

**4.3 Training**

* Train/test split: 80/20
* Evaluation: Precision, recall, F1-score
* Output: Probabilistic classification; scientists can adjust thresholds for detection.

**4.4 Deployment**

* **FastAPI:** Creates an interactive API for inference.
* **NGROK:** Allows global access for demonstration without a dedicated server.
* Can be scaled to cloud or quantum computing platforms for real-time analysis of larger datasets.

**5. Results**

* The model achieves high predictive accuracy on the test set.
* Probabilistic outputs allow confidence-based decision making:
  + Probability ≥ 0.75 → Likely planet present.
  + Probability < 0.75 → Likely no planet.
* Successfully demonstrates automation of exoplanet detection on publicly available datasets.

**6. Future Work**

* Integration with **real-time NASA telescope feeds**.
* Expansion to multi-class detection: classifying planet types (e.g., gas giant, terrestrial).
* Cloud/quantum computing deployment for TB-scale datasets.
* Incorporation of light curve analysis and computer vision models.

**7. Tools & Technologies**

|  |  |
| --- | --- |
| **Component** | **Purpose** |
| Python 3.9+ | Programming language |
| XGBoost | ML model for classification |
| SMOTE | Reduce class imbalance |
| FastAPI | Deployment framework |
| NGROK | Global API access |
| Pandas, NumPy, Scikit-learn | Data handling & preprocessing |

**8. Conclusion**

This project demonstrates how **AI can augment space science**, enabling automated detection of exoplanets with reproducible results. Using XGBoost for tabular data, probabilistic classification, and FastAPI deployment, we created a scalable solution adaptable for real-time research.

“When space telescopes see the stars, our AI helps them understand what they see.”

— *Som, Independent AI Researcher*

**9. References**

1. NASA Kepler Exoplanet Archive: https://exoplanetarchive.ipac.caltech.edu
2. Kaggle Datasets: https://www.kaggle.com
3. XGBoost Documentation: https://xgboost.readthedocs.io
4. SMOTE: Chawla et al., 2002, “SMOTE: Synthetic Minority Over-sampling Technique”
5. FastAPI: https://fastapi.tiangolo.com