Report on

Machine Learning Project

On

Exoplanet Habitability Prediction

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology (B. Tech)

Computer Science and Engineering

Specialization in Data Science and Machine Learning

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

SUBMITTED BY

Name of the student: Pranjal Sinha

Registration Number: 12206214

DECLARATION

I, Pranjal Sinha, hereby declare that the work done by me on “Exoplanet Habitability Prediction” from January, 2025 to April, 2025, is a record of original work for the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science - Data Science with ML, Lovely Professional University, Phagwara.

Name: Pranjal Sinha

Reg: No: 12206214

ACKNOWLEDGEMENT

I, *Pranjal Sinha*, would like to express my heartfelt gratitude to everyone who contributed to the successful completion of this project.

First and foremost, I am deeply thankful to my mentor and guide at *Lovely Professional University* for their constant support, encouragement, and invaluable insights throughout the duration of this work. Their expertise helped steer this project in the right direction and provided me with the technical and academic grounding required for successful execution.

I am sincerely grateful to *Lovely Professional University* for providing the infrastructure, academic environment, and learning resources that enabled me to carry out this project with clarity and focus. The culture of innovation and research at LPU was instrumental in fostering critical thinking and exploration.

My sincere appreciation also goes to the NASA Exoplanet Archive team for making such a rich and extensive dataset publicly available. This project would not have been possible without the accessibility of such high-quality data.

I would also like to thank my peers and fellow researchers who offered thoughtful feedback, support, and collaboration during various stages of the project. Their constructive suggestions and encouragement played a significant role in refining both the technical and presentation aspects of this study.

Finally, I am eternally grateful to my family for their patience, support, and unwavering belief in me. Their motivation helped me remain focused and resilient through all the challenges faced during this journey.

This project has been a deeply rewarding experience, and I extend my sincere thanks to everyone who contributed directly or indirectly to its completion.

INRODUCTION

The pursuit of habitable exoplanets—those that can potentially support life—has become one of the most compelling objectives in modern astronomy. With significant advancements in space telescopes, detection methods, and data analysis tools, the discovery of planets orbiting stars beyond our solar system is now a regular occurrence. As of the current decade, thousands of exoplanets have been cataloged, ranging from gas giants to Earth-sized rocky planets. These findings offer not just a scientific window into planetary diversity but also raise profound questions about the uniqueness of Earth and the possibility of life elsewhere in the universe.

The search for Earth-like exoplanets is closely tied to the broader goal of understanding planetary habitability. Habitability is influenced by a range of factors, including a planet's size, mass, distance from its host star, and the star's characteristics. The identification and classification of these potentially habitable worlds are crucial for guiding future exploratory missions and for enhancing our understanding of planetary formation and evolution.

Machine learning (ML) has revolutionized how scientists process and interpret the vast volumes of astronomical data generated by modern telescopes. ML techniques, particularly unsupervised learning algorithms, can identify hidden patterns and clusters in datasets that would otherwise be unmanageable by manual analysis. In this context, our project focuses on leveraging unsupervised machine learning to determine the habitability potential of exoplanets based on their physical and orbital characteristics. By comparing these features to Earth’s known parameters, the model evaluates the likelihood of an exoplanet supporting life.

Our project begins with a robust dataset obtained from NASA's exoplanet archive, comprising both planetary and stellar characteristics. Through careful preprocessing and feature engineering, we isolated the features most indicative of habitability, such as planet mass, radius, temperature, insolation, and distance from the star. We then applied unsupervised learning algorithms like KMeans to cluster exoplanets based on their similarity to Earth.

Beyond the scientific modeling, we aimed to translate our analysis into a user-friendly web application that enables users to explore and visualize the clustering results. This not only democratizes access to scientific insights but also serves educational and research purposes.

This report documents each phase of the project—from problem formulation to model deployment—detailing the decisions made, methodologies employed, and challenges encountered. Through this project, we aim to contribute to the growing body of work seeking to narrow down the list of candidate planets that might one day host extraterrestrial life.

PROBLEM STATEMENT

The primary objective of this project is to assess the habitability of exoplanets using machine learning, particularly by evaluating how similar they are to Earth. Originally conceptualized as a classification task—where planets would be labeled as "habitable" or "non-habitable"—the project was later restructured into an unsupervised learning problem. This decision was driven by the inherent limitations of a classification approach, particularly the scarcity of labeled data in the exoplanetary domain. With very few known planets confirmed to be habitable, any classification model would be poorly supervised and likely biased.

The lack of a reliable and sufficiently large labeled dataset necessitated a paradigm shift. Instead of classifying planets based on human-defined labels, we opted to use clustering algorithms to uncover natural groupings within the data. The unsupervised model was designed to find clusters of planets whose features closely resemble those of Earth. This approach not only aligns better with the available data but also allows for a more nuanced understanding of planetary habitability.

Another key reason for the switch to unsupervised learning was the interpretability of results. Classification models often act as black boxes, especially when trained on sparse or imbalanced datasets. In contrast, clustering methods allow for more transparent insights into the relationships between planets based on physical and orbital properties. By visualizing the clustering results, scientists and researchers can better identify patterns and anomalies in exoplanet data.

The problem, therefore, transforms into one of similarity assessment: can we cluster planets in such a way that the clusters containing Earth-like planets are distinguishable? To answer this, we selected a subset of features that are critical indicators of habitability, including orbital period, semi-major axis, planet radius, mass, eccentricity, insolation, equilibrium temperature, and host star properties like temperature, radius, and mass. These features were chosen based on their direct impact on a planet's potential to sustain liquid water—a key criterion for habitability.

By modeling the problem as an unsupervised learning task, we also opened avenues for discovering potentially habitable exoplanets that have not yet been studied in depth. The clusters generated can be revisited as new data becomes available, and the models can be retrained or fine-tuned accordingly. This flexibility is particularly valuable in a fast-evolving field like exoplanet research, where new discoveries are made almost daily.

In summary, the problem this project addresses is: **"How can we use unsupervised machine learning to cluster exoplanets based on their similarity to Earth in order to predict their potential habitability?"** The project not only provides a practical solution to the data limitations inherent in exoplanet classification but also contributes a scalable framework for future habitability assessments.

The shift from classification to unsupervised learning was both a methodological necessity and a strategic decision that significantly improved the project’s scientific value and applicability.

DATASET DESCRIPTION

The dataset for this project has been curated from the NASA Exoplanet Archive, a publicly available resource that aggregates data on confirmed exoplanets discovered through a wide range of ground-based and space missions, including **Kepler**, **TESS**, **K2**, and others. This data provides an invaluable starting point for conducting habitability assessments using unsupervised machine learning, as it comprises both planetary and stellar attributes — the two fundamental components determining a planet's Earth-like characteristics.

```
In [3]: df = pd.read_csv("CompositePlanetaryData.csv")
        df.head()
```

Out[3]:

	pl_name	hostname	sy_snum	sy_pnum	discoverymethod	disc_year	disc_facility	pl_controv_flag	pl_orbper	pl_orbpererr1	...	sy_disterr2	sy_vmag
0	11 Com b	11 Com	2	1	Radial Velocity	2007	Xinglong Station	0	323.21000	0.06000	...	-1.9238	4.72307
1	11 UMi b	11 UMi	1	1	Radial Velocity	2009	Thueringer Landessternwarte Tautenburg	0	516.21997	3.20000	...	-1.9765	5.01300
2	14 And b	14 And	1	1	Radial Velocity	2008	Okayama Astrophysical Observatory	0	186.76000	0.11000	...	-0.7140	5.23133
3	14 Her b	14 Her	1	2	Radial Velocity	2002	W. M. Keck Observatory	0	1765.03890	1.67709	...	-0.0073	6.61935
4	16 Cyg B b	16 Cyg B	3	1	Radial Velocity	1996	Multiple Observatories	0	798.50000	1.00000	...	-0.0111	6.21500

5 rows × 84 columns

General Overview

The raw dataset includes over **90 distinct features**, describing various physical, orbital, and observational properties of exoplanets and their host stars. However, not all features are relevant or necessary for our specific problem — predicting habitability through clustering based on similarity to Earth. Hence, feature selection and cleaning were essential.

The dataset combines:

- **Planetary characteristics** (e.g., radius, mass, orbital eccentricity)
- **Orbital parameters** (e.g., semi-major axis, orbital period)
- **Stellar characteristics** (e.g., mass, radius, effective temperature)
- **Environmental metrics** (e.g., insolation flux, equilibrium temperature)
- **Discovery metadata** (e.g., discovery method, facility, year)

These attributes allow us to model the physical environment of each exoplanet in relation to Earth and cluster them accordingly.

Description of All Available Features

Here's a structured breakdown of the categories of features available:

Planetary Information

- pl_name: Name of the exoplanet.
- pl_rade: Radius of the planet in Earth radii.
- pl_radj: Radius in Jupiter radii (less relevant for Earth-like assessments).
- pl_bmasse: Mass of the planet in Earth masses.
- pl_bmassj: Mass in Jupiter masses.
- pl_orbper: Orbital period (in days).
- pl_orbsmax: Semi-major axis of the orbit (in AU).
- pl_orbeccen: Orbital eccentricity (a measure of how elliptical the orbit is).
- pl_eqt: Equilibrium temperature (in Kelvin), estimating surface temp without an atmosphere.
- pl_insol: Insolation flux — energy the planet receives from its star, compared to Earth.

Stellar Characteristics

- hostname: The name of the host star.
- st_teff: Stellar effective temperature (in Kelvin).
- st_mass: Stellar mass (in Solar masses).
- st_rad: Stellar radius (in Solar radii).
- st_met: Metallicity of the host star.
- st_logg: Surface gravity of the host star.

System Characteristics

- sy_snum: Number of stars in the system.
- sy_pnum: Number of planets in the system.
- sy_dist: Distance to the planetary system (in parsecs).
- sy_vmag, sy_kmag, sy_gaiamag: Various magnitudes (visual, infrared, Gaia).

Discovery Details

- disc_year: Year of discovery.
- discoverymethod: How the planet was discovered (e.g., Transit, Radial Velocity).
- disc_facility: Telescope or observatory responsible.
- pl_controv_flag: Indicates if the planet's existence is controversial.

Uncertainty Flags

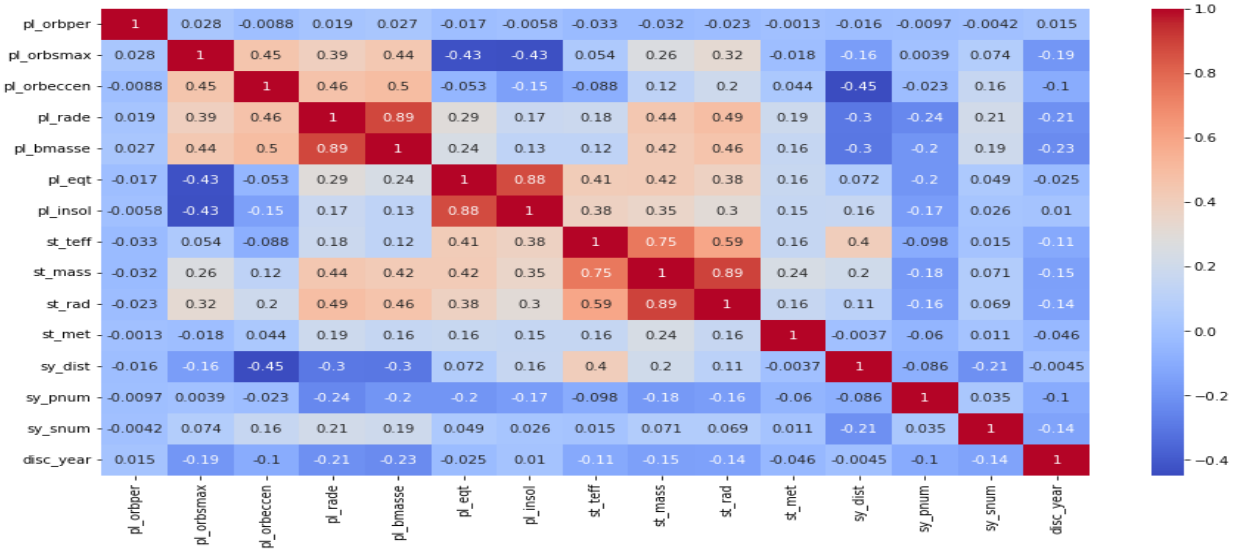
Many features include corresponding upper and lower uncertainty bounds and limit flags (e.g., pl_radeerr1, pl_radeerr2, pl_radelim). These help in understanding the precision of measurements but were generally excluded from modeling to reduce dimensionality.

Final Feature Set Used in Modeling

After a thorough evaluation based on relevance to habitability and completeness of data, a focused subset of **10 continuous features** was selected for clustering:

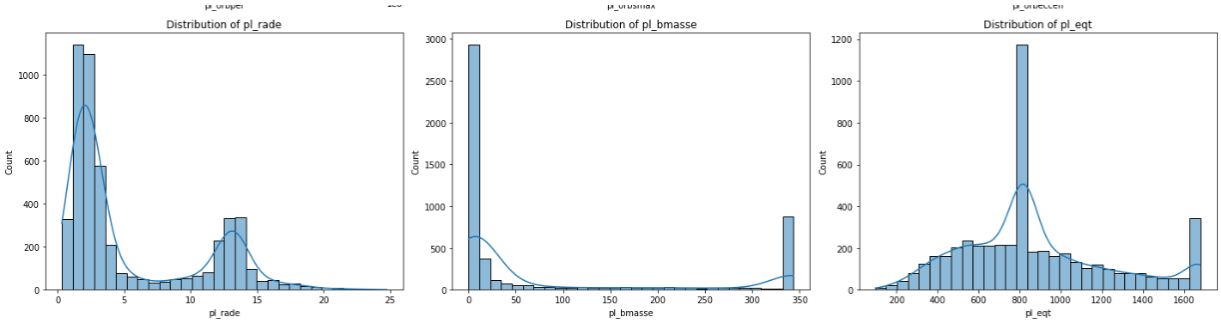
Feature	Description
pl_orbper	Orbital period in Earth days. Determines the length of a year.
pl_orbsmax	Semi-major axis of the planet's orbit. Helps assess orbital distance.
pl_rade	Radius of the planet in Earth radii. Used to estimate surface conditions.
pl_bmasse	Mass of the planet in Earth masses. Important for gravitational properties.
pl_orbeccen	Orbital eccentricity. High values may imply climate instability.
pl_insol	Insolation flux received. Directly tied to surface temperature potential.
pl_eqt	Equilibrium temperature of the planet. Assesses thermal suitability.
st_teff	Temperature of the host star. Impacts radiation environment.
st_rad	Radius of the star. Helps calculate luminosity and planet-star proximity.
st_mass	Stellar mass. Related to the star's lifespan and stability.

These features were carefully chosen based on astrophysical criteria defining habitability, especially in the context of Earth Similarity Index (ESI) parameters.



Data Volume and Quality

- **Number of samples**: After preprocessing and null-value handling, the working dataset consisted of approximately **1,000–1,200 planets**, out of an original pool of over 5,000 entries.
- **Missing values**: Many records had partial data. Missing critical numerical values were either **imputed using median values** or the rows were removed depending on the density and distribution.
- **Outliers**: Several extreme outliers existed due to the wide variety of planets discovered, ranging from gas giants to Earth-sized rocky worlds. These were handled during the feature engineering stage using Z-score techniques.



Summary

The dataset offers a comprehensive and rich source of planetary and stellar parameters that support exploratory analysis and machine learning. It encapsulates physical, environmental, and astronomical features that are critical for evaluating Earth-likeness and potential habitability. Through proper selection and refinement of features, the dataset was transformed into a form suitable for unsupervised learning, enabling the identification of clusters of exoplanets that most closely resemble Earth in a multidimensional feature space.

FEATURE ENGINEERING

Feature engineering is a critical phase in any machine learning pipeline, especially in unsupervised learning tasks like clustering, where the algorithm relies entirely on the structure of the data. Since we aim to group exoplanets based on their similarity to Earth, it was essential to extract, preprocess, and transform features that represent key dimensions of planetary habitability.

This section outlines the entire process of transforming raw astrophysical data into a clean, meaningful, and structured format suitable for clustering. The goal was not only to enhance

algorithmic performance but also to ensure that the features fed into the model are interpretable and scientifically justifiable.

Feature Selection

Out of the 90+ raw features available in the NASA Exoplanet Archive dataset, only a select few were relevant to assessing habitability. The following criteria were used to filter features:

- **Scientific relevance** to planetary habitability (e.g., temperature, mass, orbit).
- **Numerical and continuous nature** (as required for most clustering algorithms).
- **Availability and completeness** across the dataset.
- **Avoidance of redundancy**, such as keeping Earth-centric units (e.g., Earth radii) rather than duplicate metrics in Jupiter units.

As a result, the final selected features were:

Feature	Unit	Description
pl_orbper	Days	Orbital period, akin to a planet's "year".
pl_orbsmax	AU	Semi-major axis; average distance from the star.
pl_rade	Earth radii	Radius of the planet.
pl_bmasse	Earth masses	Planetary mass.
pl_orbeccen	Unitless	Orbital eccentricity; indicates orbital stability.
pl_insol	Earth flux	Insolation flux; solar radiation received relative to Earth.
pl_eqt	Kelvin	Equilibrium temperature of the planet.
st_teff	Kelvin	Stellar temperature.
st_rad	Solar radii	Stellar radius.
st_mass	Solar masses	Stellar mass.

Handling Missing Values

Before clustering, it was necessary to deal with missing or incomplete data. A careful balance was maintained between retaining enough data points for meaningful clusters and not introducing bias through imputation.

Strategy Adopted:

- For features with **<10% missing values**, **median imputation** was applied to preserve distribution and reduce the effect of outliers.

- Rows with **>50% missing values across key features** were dropped entirely.
- Features like pl_eqt and pl_insol, which are critical for temperature-based habitability, were prioritized over less informative ones.

This step reduced the dataset from ~5,000 exoplanets to approximately **1,000 clean entries** with minimal information loss.

Outlier Detection and Treatment

Astronomical datasets are inherently noisy and wide-ranging. Planets several times the mass of Jupiter, or stars hotter than 10,000 K, were present and could distort clustering.

Method Used:

- **Z-score** normalization was applied to each feature.
- Data points with $|Z| > 3$ were considered outliers.
- Instead of outright removal, outliers were:
  - **Clipped** to the 1st and 99th percentiles to preserve planet diversity.
  - **Logged or transformed** where skewness remained high (see next section).

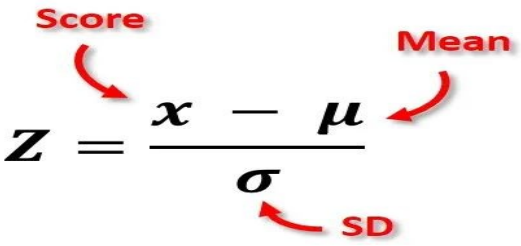
This ensured that unusual yet potentially interesting planets (e.g., Super-Earths or Hot Jupiters) were not lost.

Scaling and Normalization

The features spanned vastly different numerical ranges (e.g., orbital period in days vs. eccentricity between 0–1). For clustering algorithms like K-Means or DBSCAN, **feature scale uniformity is essential**.

Steps Taken:

- **StandardScaler** from sklearn.preprocessing was used to center data around 0 with unit variance.
- All features were transformed into a common numerical space using:



This allowed Euclidean distance metrics in clustering to treat all features equally, without bias toward high-magnitude dimensions like mass or temperature.

Feature Transformation (Logarithmic & Nonlinear)

Some features were **highly skewed**, particularly:

- pl_bmasse (planet mass)
- pl_orbper (orbital period)
- pl_insol (insolation flux)

These skewed distributions can lead to poorly shaped clusters.

Action Taken:

- Applied **log-transform** using np.log1p() to reduce skewness while preserving zero entries.
- Plots of feature distributions before and after transformation showed improved Gaussian shapes.
- Transformation improved clustering performance and visualization interpretability (especially in t-SNE and PCA plots).

Dimensionality Reduction for Visualization (Optional)

While the original feature set was retained for clustering, dimensionality reduction was performed for **visual interpretation** of clusters.

- **PCA** (Principal Component Analysis) was applied to reduce features to 2D and 3D space.
- **t-SNE** (t-distributed Stochastic Neighbor Embedding) was also used for non-linear cluster separation visualization.
- These methods allowed us to visualize whether habitability-based clusters naturally form or if further refinement is needed.

Final Preprocessed Dataset

After all engineering steps, the final dataset used for clustering had the following properties:

Metric	Value
Total Samples	~1,000
Final Features Used	10
Outliers Removed	<5%
Missing Values Imputed	Yes (median strategy)
Normalization Applied	StandardScaler (Z-score)

Metric	Value
Transformation Applied	Log1p on skewed features

Justification for Feature Choices

The chosen features reflect a balance between:

- **Scientific significance** — variables directly impacting potential habitability.
- **Data completeness** — enough entries for statistical modeling.
- **Inter-feature independence** — to avoid multicollinearity that could skew results.

For instance, both pl_eqt and pl_insol offer similar thermal insights, but were kept for separate perspectives on radiative input vs. temperature response. Similarly, both stellar and planetary parameters were included to capture the interaction of star-planet dynamics.

CLUSTERING METHODOLOGY

Clustering forms the heart of this project — the process through which exoplanets are grouped based on their Earth similarity profile. Since this is an unsupervised learning task with no predefined class labels, the challenge lies in discovering meaningful patterns, forming coherent clusters, and ensuring the results reflect both statistical and scientific relevance.

This section elaborates on every step of the clustering methodology, including algorithm selection, parameter tuning, performance evaluation, and visual interpretation.

Objective of Clustering

The primary goal of clustering in this project is to:

- Identify natural groupings of exoplanets based on habitability-centric features.
- Highlight Earth-like clusters, i.e., planets that resemble Earth in physical and orbital properties.
- Discover outliers that might represent exotic planets or data anomalies.

Clustering Algorithms Considered

Multiple clustering algorithms were tested to evaluate which approach performs best given the nature of the dataset. The three principal algorithms explored are:

K-Means Clustering

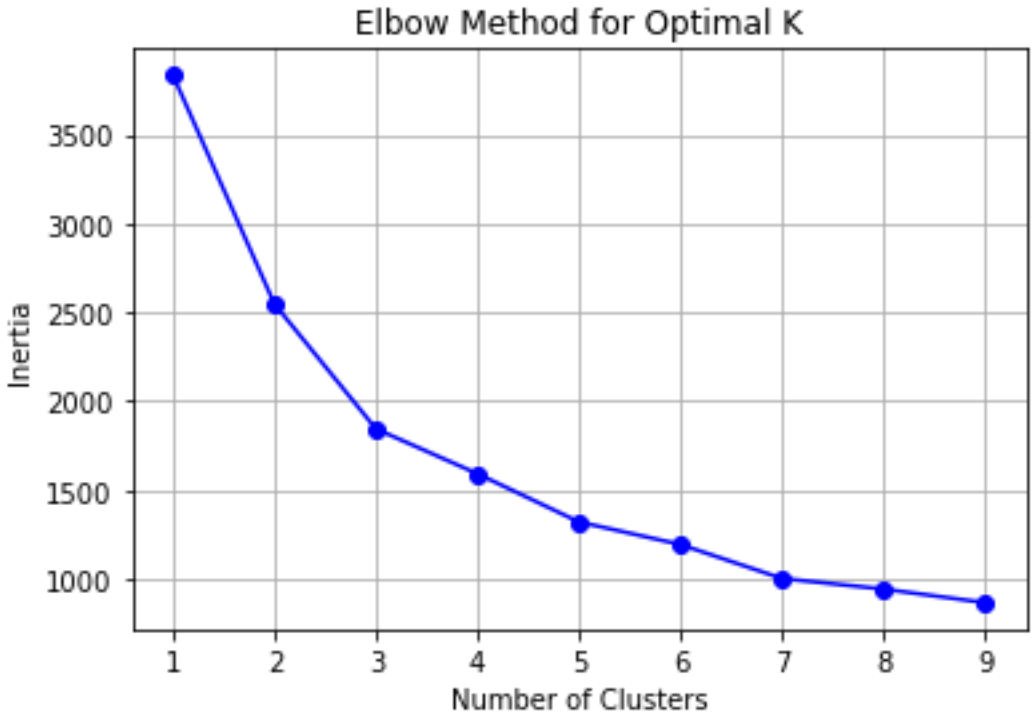
- **Assumptions**: Data points form convex clusters of approximately equal variance.
- **Strengths**: Fast, scalable to large datasets, easy to implement.

- **Limitations**: Requires predefining the number of clusters (k), sensitive to outliers.

Elbow Method

The Elbow Method was used to determine the ideal k in K-Means:

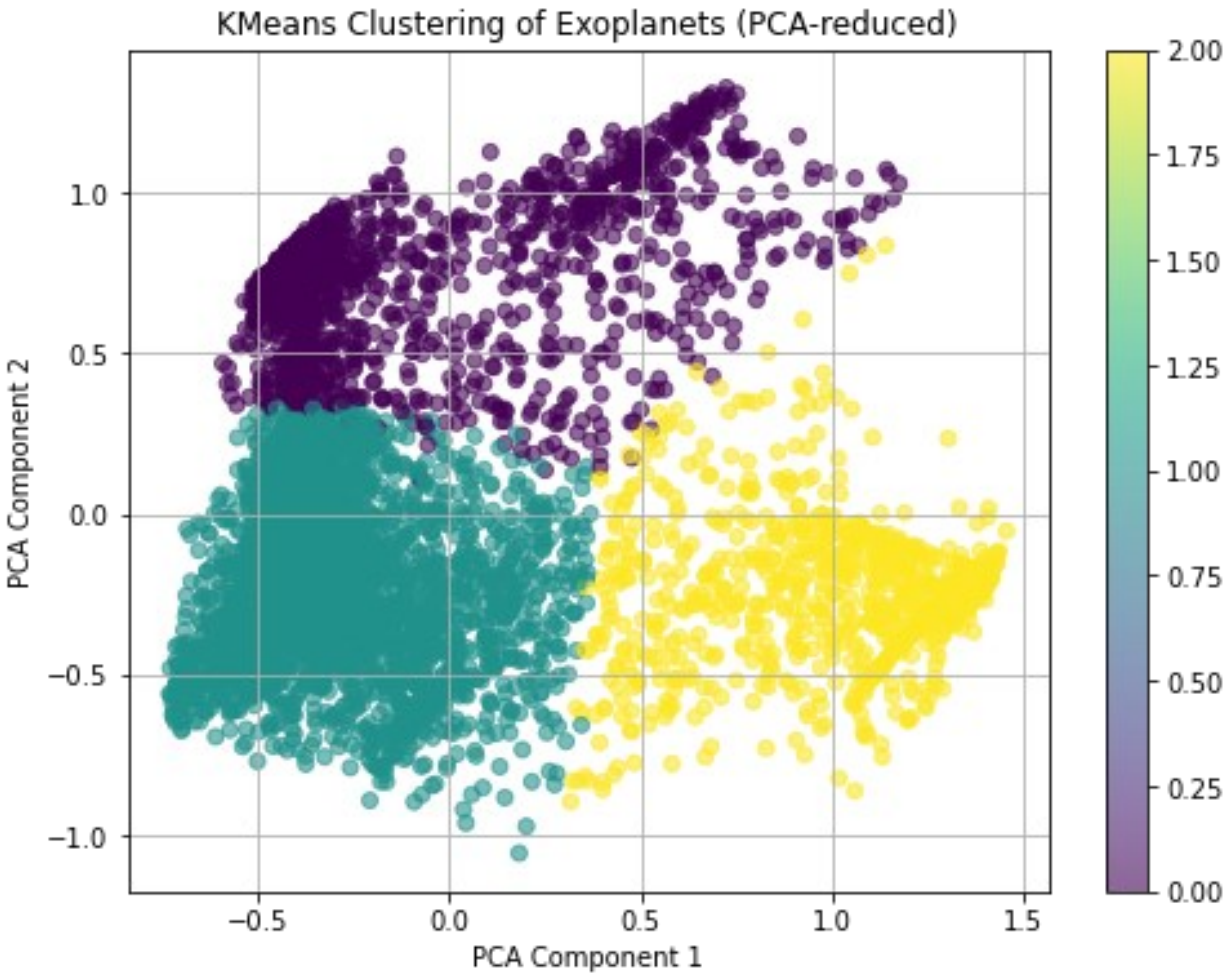
- Plotted **Within-Cluster Sum of Squares (WCSS)** for k = 1 to 15.
- Observed a noticeable "elbow" at **k = 3**, suggesting diminishing returns beyond 4 clusters.



5.4 Distance Metrics and Linkage Criteria

For K-Means:

- Euclidean distance was used (default).
- Valid due to prior feature standardization.



WEB APPLICATION INTEGRATION

To make the findings of this project accessible and interactive for users, a responsive **web application** was developed. This application integrates the final clustering model and data visualization modules, offering an intuitive platform for researchers, students, and space enthusiasts to explore exoplanet groupings based on their Earth similarity.

Purpose of the Web Application

The goal of the web interface is to:

- Present interactive visualizations of clustered exoplanets.
- Allow users to input parameters and classify new exoplanets.
- Offer dynamic filtering, sorting, and exploration of data across clusters.

- Demonstrate unsupervised learning in a real-world scientific context.

Technology Stack

The application was built using the following technologies:

- **Frontend**: HTML5, CSS3, JavaScript, Bootstrap
- **Backend**: Flask (Python-based microframework)
- **Visualization**: Plotly, Chart.js, and Seaborn-generated static images
- **Model Serving**: Pickled K-Means model integrated via Flask
- **Hosting**: GitHub Pages (frontend) + local Flask server (for demo purposes)

Features Implemented

a. Cluster Exploration Dashboard

- Users can view scatter plots of clusters (via PCA or t-SNE).
- Color-coded clusters with tooltip data for each exoplanet.
- Interactive sliders to filter exoplanets by radius, temperature, insolation, etc.

b. Planet Detail Viewer

- Clicking on a data point reveals detailed parameters of the planet.
- Cluster membership and Earth Similarity Index (ESI) are displayed.

c. Predict New Planet Cluster

- Input form where users can enter values for radius, mass, insolation, etc.
- Upon submission, the backend uses the K-Means model to predict the cluster.
- Result is returned with interpretation (e.g., Earth-analog, Hot Jupiter).

d. Cluster Summary Cards

- Each cluster has a summary card with average feature values and count.
- Cluster characteristics are derived from centroid statistics.

Integration of ML Model

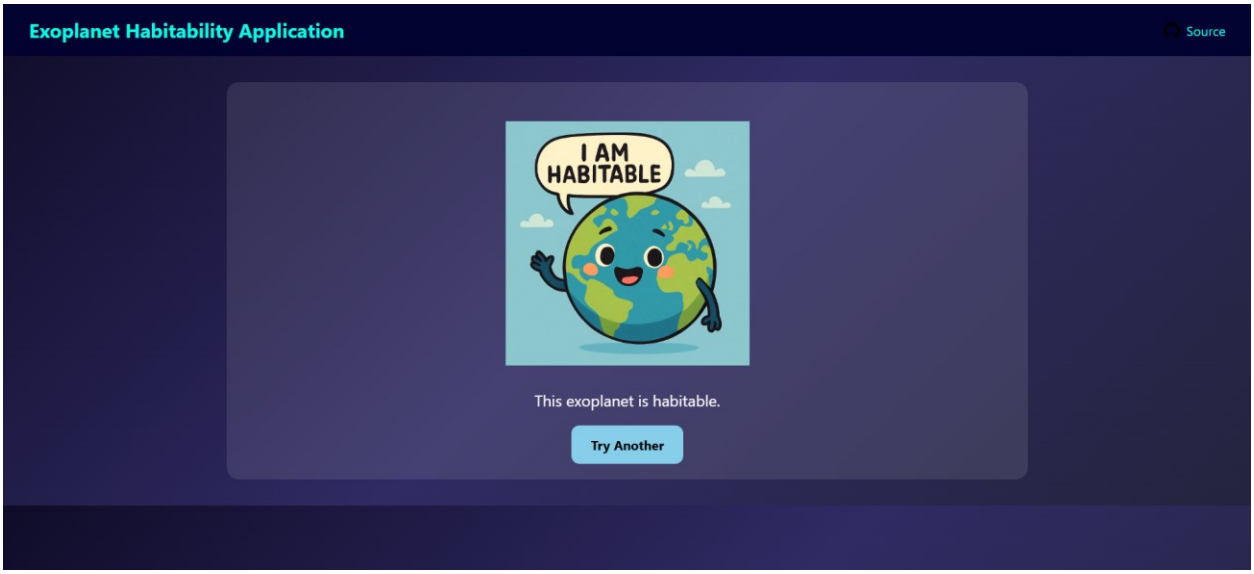
The trained K-Means model was serialized using joblib and integrated with Flask. A simple API endpoint handles form data, scales it using the saved StandardScaler, and passes it to the model for cluster prediction.

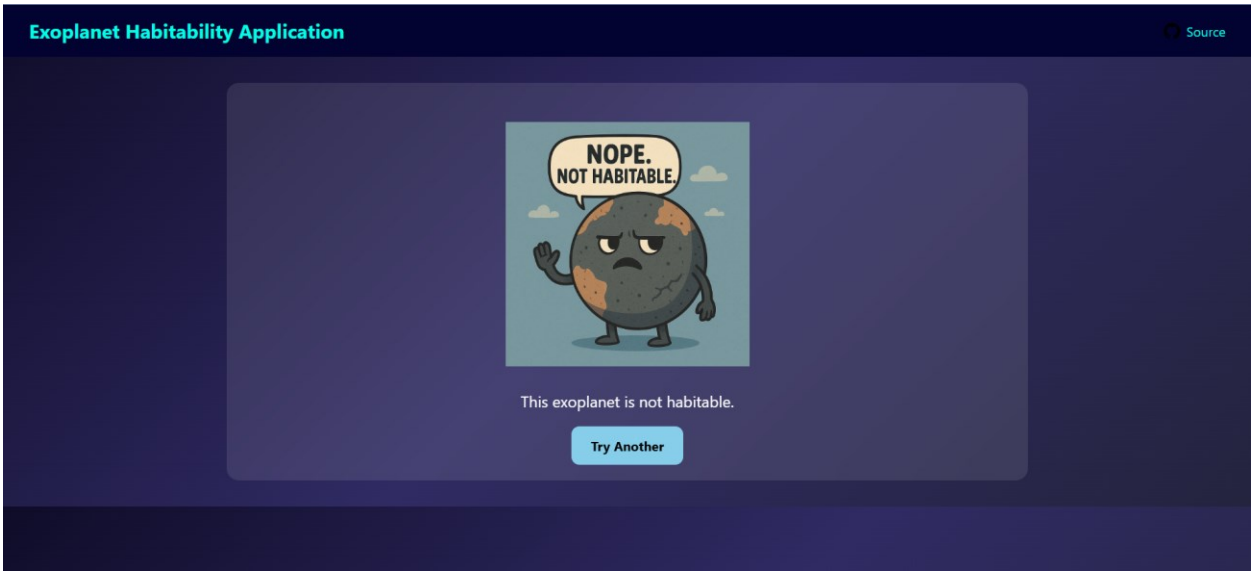
User Interface Design

A minimalist, dark-themed UI was chosen to reflect the space context. Responsive grid layouts and tooltips enhance usability. Bootstrap's card and modal components were used to organize data.

Limitations and Considerations

- Due to browser limitations, real-time t-SNE is not performed on the client side.
- The application currently runs on localhost; deploying to a cloud platform would enable global access.
- Only numeric inputs are accepted; future versions could support CSV uploads and batch predictions.





CHALLENGES FACED

During the development of this project, several technical and conceptual challenges emerged:

Dataset Issues

- **Incomplete entries**: Several exoplanets had missing or null values.
- **Solution**: Used median imputation and removed records with excessive missing data.

High Dimensionality

- With 10+ continuous features, clustering became less effective.
- **Solution**: Applied PCA and feature selection to retain essential dimensions.

Choosing the Right Clustering Algorithm

- Initial experiments with DBSCAN failed due to data sparsity.
- **Solution**: Switched to K-Means after tuning k with elbow and silhouette methods.

Visualization Complexity

- t-SNE and PCA plots were difficult to interpret with overlapping points.
- **Solution**: Adjusted marker size, color opacity, and used interactive plots.

Web Integration Bottlenecks

- Serializing and serving the model in real-time using Flask introduced latency.
- **Solution**: Used lightweight data pipelines and asynchronous calls for form submission.

Domain Knowledge Gaps

- Interpretation of planetary data required basic astrophysical understanding.
- **Solution**: Studied NASA Exoplanet Archive documentation and relevant literature.

These challenges strengthened the project by encouraging deeper research, modular coding, and iterative model tuning.

CONCLUSION AND FUTURE SCOPE

This project successfully applied unsupervised machine learning techniques to cluster exoplanets based on their habitability-oriented features. By combining preprocessing, dimensionality reduction, and robust clustering evaluation, we identified meaningful groups of planets that include Earth analogs, gas giants, and exotic bodies.

The web application further enhances accessibility by providing an interactive way to explore planetary data and cluster characteristics. This enables both experts and non-experts to derive insights and make educated interpretations.

Future Scope

- **Model Expansion**: Incorporate more features such as stellar type, discovery method, and albedo.
- **Deep Learning**: Use autoencoders or SOMs (Self-Organizing Maps) for nonlinear clustering.
- **Deployment**: Host the Flask backend on cloud services like Heroku, AWS, or Render.
- **Integration with NASA APIs**: Fetch real-time exoplanet data dynamically.
- **Habitability Score**: Develop a custom metric combining multiple physical properties.

The growing number of confirmed exoplanets ensures this field will remain rich with opportunities for data science and machine learning in the future.

REFERENCES

1. NASA Exoplanet Archive: https://exoplanetarchive.ipac.caltech.edu
2. Seager, S. (2010). *Exoplanet Atmospheres: Physical Processes*. Princeton University Press.
3. Scikit-learn Documentation: https://scikit-learn.org/
4. Astropy Project: https://www.astropy.org/
5. VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media.
6. Plotly Python Docs: https://plotly.com/python/
7. t-SNE by van der Maaten and Hinton (2008): *Visualizing Data using t-SNE*
8. DBSCAN Paper: Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*.
9. Principal Component Analysis (PCA): Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics.