**Data-Driven Astronomy Project Plan**

**Project Title:**

Exploring Planetary Data with Visualization, Predictive Modeling, and Interactive Elements

**Team Members:**

1. [Name 1]
2. [Name 2]
3. [Name 3]

**Project Objectives:**

* Analyze planetary datasets to uncover insights about planetary properties.
* Build data visualizations to explore relationships between planetary attributes.
* Develop predictive models to forecast planetary characteristics.
* Implement interactive elements for user-driven data exploration.
* Create a user-friendly GUI.

**Technology Stack:**

* **Data Handling:** Python (Pandas)
* **Data Visualization:** Matplotlib, Seaborn, Plotly
* **Predictive Modeling:** Scikit-learn
* **Interactive GUI:** Dash or Tkinter
* **Version Control:** GitHub
* **Documentation:** Markdown files, Jupyter Notebooks
* **Development Environment:** VS Code

**Project Timeline:**

**Day 1: Initial Project Setup**

* **Kick-off Meeting:**
  + Define project scope and goals.
  + Agree on team member responsibilities.
  + Discuss collaboration tools.
* **Development Environment Setup:**
  + Install Python, VS Code, and GitHub Desktop.
  + Create a shared GitHub repository.
  + Set up the following project structure:
  + project\_directory/
  + ├── data/ # for datasets
  + ├── notebooks/ # for exploratory analysis
  + ├── src/ # for code scripts
  + ├── docs/ # for documentation
  + └── README.md

**Day 2-3: Data Gathering & Cleaning**

* **Data Collection:**
  + Assign one member to gather data from reliable sources (e.g., NASA Exoplanet Archive).
  + Document each data source in docs/data\_sources.md.
* **Data Cleaning:**
  + Handle missing values and outliers.
  + Normalize or transform data as needed.
* **Documentation:**
  + Maintain a data\_cleaning\_log.md file to record cleaning steps and justifications.

**Day 4-6: Exploratory Data Analysis (EDA)**

* **Visualization Tasks:**
  + Member 1: Visualize size and mass comparisons.
  + Member 2: Analyze temperature and distance relationships.
  + Member 3: Examine correlations between planetary properties.
* **Tools:** Use Matplotlib and Seaborn.
* **Documentation:**
  + Record findings and insights in docs/eda\_report.md.

**Day 7-10: Predictive Modeling**

* **Model Development:**
  + Select features and target variables.
  + Experiment with models (e.g., Linear Regression, Decision Trees).
  + Evaluate model performance.
* **Documentation:**
  + Maintain a modeling\_log.md file to document model choices, parameters, and evaluation metrics.

**Day 11-14: Interactive User Interface**

* **GUI Development:**
  + Use Dash for building the interactive interface.
  + Tasks:
    - Member 1: Build data filters.
    - Member 2: Integrate plots.
    - Member 3: Manage user interactions and layout.
* **Testing:** Ensure the interface is user-friendly and responsive.
* **Documentation:**
  + Create docs/gui\_guide.md for usage instructions.

**Day 15-17: Final Documentation & Presentation**

* **Documentation:**
  + Write a comprehensive README.md file with:
    - Project overview
    - Installation instructions
    - Usage guide
  + Organize technical documentation in /docs directory.
* **Testing & Debugging:**
  + Ensure the project runs smoothly on different systems.
* **Presentation:**
  + Each member presents a part of the project.
  + Include visualizations and model insights.

**Project Management Guidelines:**

* **Daily Check-ins:** 15-minute stand-up meetings to share progress and blockers.
* **Version Control:**
  + Commit changes regularly.
  + Write clear commit messages.
* **Branching:** Create separate branches for features and merge them carefully.
* **Code Reviews:** Peer-review each other's work for learning and improvement.

**Additional Notes:**

* **Documentation Best Practices:**
  + Document while progressing through the project.
  + Use Jupyter Notebooks for exploratory analysis documentation.
* **GitHub Usage:**
  + Avoid "Add README" option during repository creation if a README file already exists locally.
* **Data Handling:**
  + Handle large datasets in Python for better efficiency.
* **Learning Opportunities:**
  + Explore advanced visualization techniques.
  + Learn best practices for predictive modeling and hyperparameter tuning.
  + Familiarize yourself with GUI development using Dash.