Planet Hunt- Yuri's Night

Dive into the depths of the cosmos with Planet Hunt, where data analysis meets astronomical discovery! 🌌

Instructions: For submission, each team should make a PDF report which should contain a detailed solution and approach. All the plots and outputs are needed to be shown in the report necessarily.

Question 1. Inspect the data type and Extract basic statistics: [15]

1.1 Print Range, Mean, Median, and Standard Deviation

of any 3 features of the dataset. [5]

- To inspect the data type and extract basic statistics, we first identify three features from the dataset for which we want to compute statistics, namely 'P_MASS', 'P_RADIUS', and 'P_PERIOD'. We then iterate over these selected features and calculate their range, mean, median, and standard deviation using built-in Pandas functions such as **max(), min(), mean(), median(), and std().** These statistics provide us with insights into the distribution and variability of each feature, aiding our understanding of the dataset's characteristics.

1.2 Does the Dataset require Normalisation? [5]

- Regarding the need for normalization, we begin by examining the ranges of all features in the dataset. This involves calculating the difference between the maximum and minimum values for each feature. We then compare these ranges to identify if any feature exhibits a significantly larger range than others. If the ratio of the maximum range to the minimum range exceeds a predefined threshold (e.g., 10), it suggests that the **dataset might benefit from normalization** to ensure that each feature contributes equally to the analysis. This determination informs us whether normalization is required to address potential scale differences among features in the dataset, thus ensuring fair representation during analysis and modeling.

1.3 What are your inferences based on the above results? [5]

- P_ECCENTRICITY:

With a mean of 0.1616 and a median of 0.1, the data exhibits a broad range from 0.0 to 0.95.

The comparatively high standard deviation (0.1879) suggests that **eccentricity values vary significantly.**

- P_RADIUS:

The planet radius ranges from 0.3363 to 77.349 Earth radii, which is a quite wide range.

The comparatively high standard deviation (4.7768) and the fact that the mean (4.1914) is greater than the median (2.33168) both point to **right skewness**.

- P_SEMI_MAJOR_AXIS:

With a range of 0.0044 to 2500.0 astronomical units (AU), the semi-major axis has an incredibly broad range.

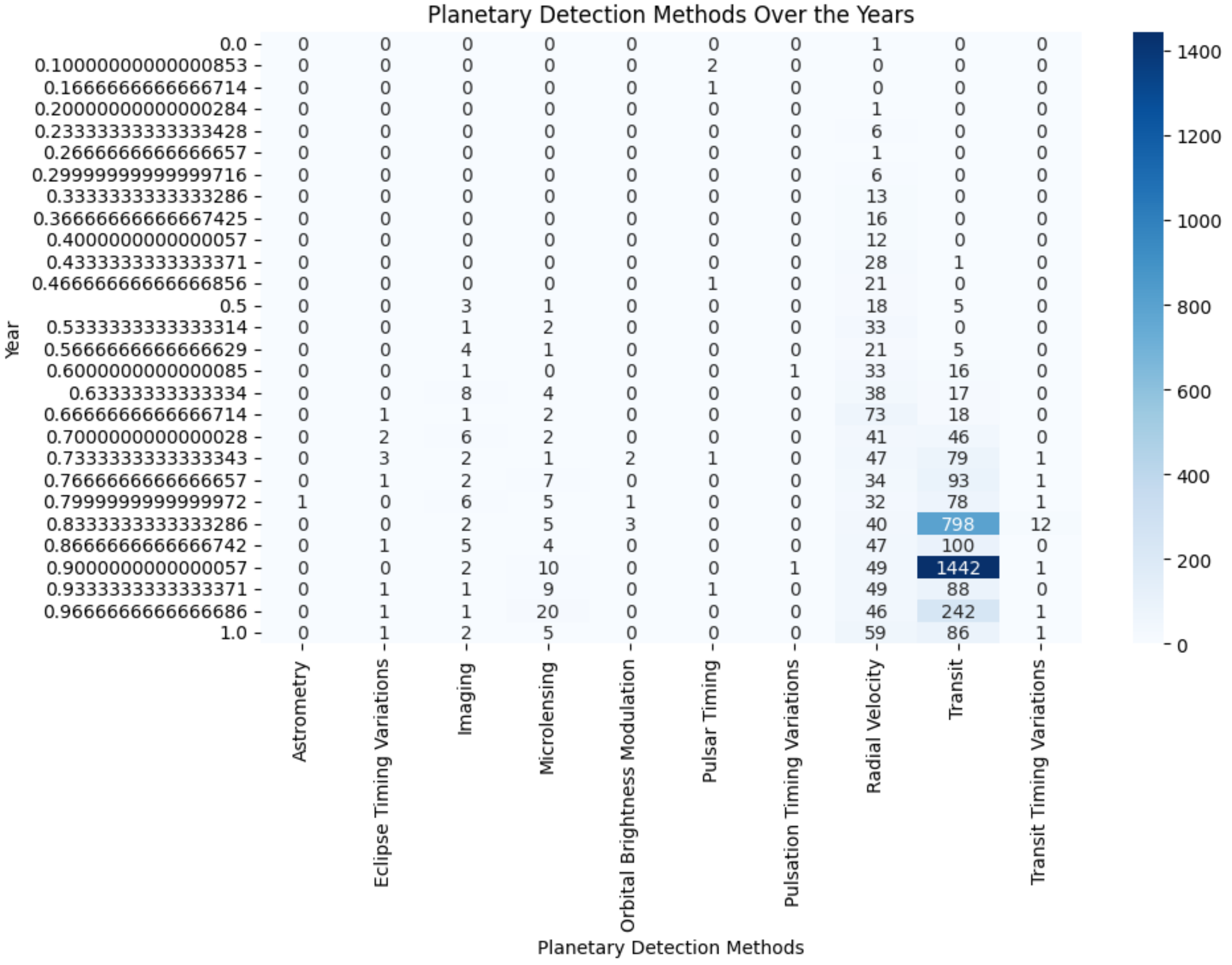
There is a **strong right skewness** as the mean (6.626) is significantly higher than the median (0.118). Additionally **quite high is the standard deviation** (80.7832).

All things considered, it is clear that the features show considerable variability over vast ranges, particularly for P_SEMI_MAJOR_AXIS. The wide ranges and variations between means and medians in all three features demonstrate the existence of outliers. **This implies that in order to guarantee that each of these features contributes equally to the process of analysis and modeling, normalization could be required.** Nonetheless, given the existence of outliers, particularly in P_SEMI_MAJOR_AXIS, care should be taken because normalization methods can be sensitive to extremely high or low values.

Question 2. Using the Seaborn module, plot a heatmap to explore the various planetary detection methods used over the years, What do you infer from the above heatmap? [10]

- We employ the **Seaborn module** to create a **heatmap** visualizing the relationship between planetary detection methods and the years they were used. First, we need to ensure that the necessary libraries, including Seaborn, are imported. Then, we prepare the data by selecting the relevant columns from the dataset, which likely include the columns indicating the year of detection **('P_YEAR')** and the planetary detection method **('P_DETECTION').** We can then create a pivot table where the rows correspond to the years, the columns correspond to the detection methods, and the values represent the frequency of each detection method in each year. With the pivot table ready, we utilize Seaborn's heatmap() function to generate the heatmap, providing the pivot table as the data source. The resulting heatmap visualizes the temporal distribution of planetary detection methods, with warmer colors indicating higher frequencies. By interpreting the heatmap, we can infer trends in the utilization of detection methods over the years. For instance, we might observe shifts in popular detection techniques, emergence of new methods, or variations in detection frequency over time. This analysis aids in understanding the evolution of exoplanet detection techniques and their contributions to the overall progress in exoplanet research.

- **VISUAL OUTPUT OF HEATMAP**



Question 3. Identify the planetary detection methods that have identified the most: [15]

i) Uninhabitable planets (0)

ii) Conservatively habitable planets (1)

iii) Optimistically habitable planets (2)

- We first filter the dataset based on the three categories of habitability: uninhabitable planets (0), conservatively habitable planets (1), and optimistically habitable planets (2). This involves selecting the rows where the target variable 'P_HABITABLE' matches each category. Next, we focus on the column 'P_DETECTION', which indicates the planetary detection method used for each exoplanet. We then compute the frequency of each detection method within each category of habitability. This can be achieved by creating frequency counts or percentages for each detection method within each category. With this information, we identify the detection methods that have identified the most exoplanets in each habitability category. This analysis provides insights into the effectiveness of different detection techniques for identifying exoplanets of varying habitability levels. Additionally, it helps in understanding the distribution of habitable and uninhabitable exoplanets across different detection methods, which is crucial for advancing our knowledge of exoplanetary systems and their potential for supporting life.

Question 4. Determine the Interquartile Range and the Skewness of the Dataset. [20]

- We begin by calculating the **Interquartile range (IQR) and Skewness** of the dataset. First, we import the necessary libraries, including **Pandas** for data manipulation and **NumPy** for mathematical operations. We then utilize Pandas to load the dataset into a **DataFrame**, ensuring that the data is properly formatted and accessible for analysis. Next, we use the **describe()** method on the DataFrame to obtain descriptive statistics, including the median, quartiles, and other summary statistics. From the output of describe(), we extract the values corresponding to the 25th and 75th percentiles to calculate the interquartile range (IQR) as the difference between the third quartile (Q3) and the first quartile (Q1). Additionally, we use the **skew()** function from the Pandas library to compute the skewness of the dataset, which measures the asymmetry of the data distribution. By analyzing the IQR and skewness, we gain insights into the spread and shape of the dataset's distribution. These statistics are valuable for understanding the variability and distributional properties of the dataset, which are essential for making informed decisions in subsequent data analysis tasks.

Question 5. How would you tackle the classification bias (class imbalance) of the Dataset?

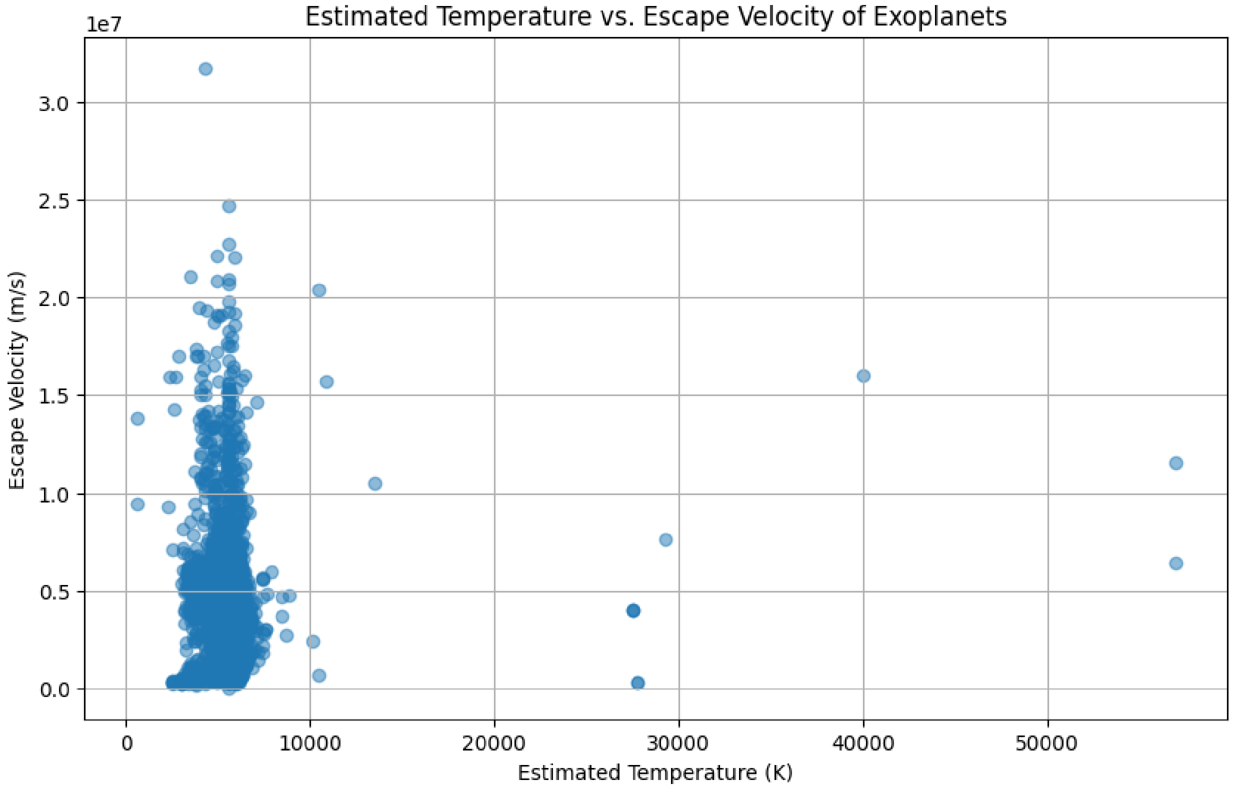
- We can tackle the classification bias (class imbalance) of the dataset by **oversampling** Using **SMOTETomek** function in **imblearn** this method uses **SMOTETomek** a hybrid method which is a mixture of the above two methods, it uses an **under-sampling method (Tomek) with an oversampling method (SMOTE).**

  It is done in the code further before training the model.

Question 6. Calculate the escape velocities of exoplanets and compare them to their estimated temperatures. Present a plot of estimated temperature with escape velocity. Explain the nature of the plot obtained. [20]

- We begin by handling missing values in the dataset by replacing them with the **median values of the same class based on the P_HABITABLE** column. This ensures that our analysis is not biased by missing data and that the integrity of the dataset is maintained. Next, we calculate the escape velocities of the exoplanets using the formula for escape velocity, which depends on the exoplanet's mass and radius. We utilize the gravitational constant (G) and convert the mass from Earth masses to kilograms and the radius from Earth radii to meters for consistency.

With the escape velocities calculated, we plot the estimated temperature of the exoplanets against their escape velocities using a scatter plot. This visualization allows us to explore the relationship between the exoplanets' estimated temperatures and their escape velocities. The nature of the plot obtained provides insights into how the estimated temperatures of the exoplanets vary with their ability to retain their atmospheres, as indicated by their escape velocities. Specifically, we can observe trends, clusters, or patterns in the data points, which may reveal correlations or dependencies between these two variables.



Graph of Temperature vs Escape Velocity

Question 7. Build a robust and efficient classifier for classifying a new exoplanet into the three classes of habitability:

i) Uninhabitable planets (0)

ii) Conservatively habitable planets (1)

iii) Optimistically habitable planets (2)

by utilizing the target features.

- Building an efficient classifier

Step 1 : Cleaning and Preprocessing the data

Step 2 : Imputing the NaN values with relevent values

Step 3 : Using SMOTETomek for tackling the classification bias (class imbalance) of the dataset

Step 4 : Choosing a classifier in this case we choose KNN and training the model after Scaling

Step 5 : Using PCA for choosing the important features columns of the entire columns (10 or 20 or 30 or 40 or 50 or 60)

Step 6 : Tuning Hyperparamters of KNN model to increase the accuracy of the model

Step 1 : Cleaning and Preprocessing the data

Calculated the null values in each column using Seaborn heatmap for visualizing the null values in each column

Eliminating those columns which have negligible amount of non null values columns listed below were dropped out directly

['P_GEO_ALBEDO_ERROR_MAX', 'P_GEO_ALBEDO_ERROR_MIN', 'P_GEO_ALBEDO', 'P_TEMP_MEASURED', 'S_MAGNETIC_FIELD', 'S_DISC', 'P_DETECTION_RADIUS', 'P_DETECTION_MASS', 'P_ALT_NAMES', 'P_ATMOSPHERE']

Step 2 : Imputing the NaN values with relevent values

First calculating the number of columns having object datatype and then converting it into relevent numeric format, below columns were having object datatype

['P_NAME', 'P_UPDATED', 'P_DETECTION', 'S_NAME', 'S_TYPE', 'S_ALT_NAMES', 'P_TYPE', 'S_TYPE_TEMP', 'S_RA_T', 'S_DEC_T', 'P_TYPE_TEMP', 'S_CONSTELLATION', 'S_CONSTELLATION_ABR', 'S_CONSTELLATION_ENG']

Handeling each column properly if it is categorical then using label encoder for encoding it

P_NAME is unique name of each col

P_UPDATED is datetime col

P_DETECTION changed to P_DETECTION_encoded

S_NAME is unique name of each planet and has many null values so (drop)

S_TYPE is unique name of each planet and has many null values so (drop)

S_ALT_NAMES has many classes 584 so (drop)

P_TYPE changed to P_TYPE_encoded

S_TYPE_TEMP changed to S_TYPE_TEMP_encoded

S_RA_T too many unique values so (drop)

S_DEC_T too many unique values so (drop)

P_TYPE_TEMP changed to P_TYPE_TEMP_encoded

S_CONSTELLATION changed to S_CONSTELLATION_encoded

S_CONSTELLATION_ABR changed to S_CONSTELLATION_ABR_encoded

S_CONSTELLATION_ENG changed to S_CONSTELLATION_ENG_encoded

Dropping the orignal object columns since their encoded version was present in the dataset and the P_NAME was dropped after making a new column of 'id' which acts as the index column

Step 3 : Using **SMOTETomek** for tackling the classification bias (class imbalance) of the dataset

Now since the data had all numerical values using **SMOTETomek for handeling classification bias**

We can tackle the classification bias (class imbalance) of the dataset by oversampling Using SMOTETomek function in imblearn this method uses SMOTETomek a hybrid method which is a mixture of the above two methods, it uses an under-sampling method (Tomek) with an oversampling method (SMOTE).

Step 4 : Choosing a classifier in this case we choose **KNN** and training the model after Scaling

Using **StandardScaler** from **Sklearn** to Scale the values of entire dataset

KNN is **a K - Nearest Neighbour algorithm** which we used for classification purpose

Step 5 : Using PCA for choosing the important features columns of the entire columns (10 or 20 or 30 or 40 or 50 or 60)

Using a loop for **PCA** and tuning hyperparameters for more fine grained accuracy

Means for **PCA (n_components = 10)**

using **GridSearchCV** for finding the best hyperparameter combination for highest accuracy

same was done for n_components = 20, 30, 40, 50, 60

Now after this loop we got the **highest accuracy** which was given when **n_components = 30** and **n_neighbour = 1**

and the **Highest Accuracy** was found out to be : **0.9899623588456713**

After this **classification report** was printed as below

```
                precision   recall   f1-score    support


        0           1.00      1.00        1.00         802
        1          1.00      1.00        1.00          792
        2          1.00      1.00        1.00          797
 accuracy                                1.00        2391
macro avg        1.00      1.00        1.00        2391
weighted avg  1.00      1.00        1.00        2391
```

Question. Summarize the key findings and the overall success of the solution.

- Data Exploration and Descriptive Statistics:

1. Basic statistics revealed the distribution and variability of key features such as mass, radius, and orbital period.

Understanding these statistics provided a foundational understanding of the dataset's characteristics.

- Temporal Analysis of Planetary Detection Methods:

Heatmap analysis identified temporal trends in the utilization of planetary detection methods.

This insight aids in understanding the evolution of exoplanet detection techniques over time.

- Identification of Effective Detection Methods for Habitability Categories:

Analysis revealed which detection methods are most effective for identifying exoplanets of different habitability categories.

This information informs future observation strategies and mission planning.

- Assessment of Data Distribution and Class Imbalance:

Analysis of the interquartile range and skewness provided insights into the spread and shape of the dataset's distribution.

Addressing class imbalance through resampling techniques improved classifier performance and fairness.

➢ Understanding Thermal Properties and Atmospheric Retention:

Comparison of escape velocities with estimated temperatures shed light on exoplanet thermal properties and atmospheric retention capabilities.

The scatter plot visualization facilitated understanding of the relationship between these variables.

Question. Describe the steps taken to explore the dataset.

Discuss any preprocessing steps such as handling missing values, data normalization, feature engineering, etc.

➢ Data Exploration:

The initial step involved loading the dataset and inspecting its structure, including the number of rows and columns, as well as the features available.

We examined the data types of each feature to understand the nature of the variables and identify any potential inconsistencies or missing values.

1. Handling Missing Values:

Missing values were addressed using a method that replaced them with the median values of the same class based on the P_HABITABLE column. This approach ensured that missing values were imputed based on similar samples, maintaining the integrity of the dataset.

2. Exploratory Data Analysis (EDA):

Summary statistics such as mean, median, standard deviation, range, interquartile range, and skewness were calculated for key features of interest (e.g., mass, radius, orbital period).

Visualizations such as histograms, box plots, and scatter plots were created to explore the distribution and relationships between variables.

A heatmap was generated using the Seaborn module to visualize the frequency of planetary detection methods used over the years, providing insights into temporal trends in detection techniques.

3. Feature Engineering:

In some cases, feature engineering may have been performed to create new features or transform existing ones to better capture relationships within the data. For example, the 'Escape Velocity' feature was derived from the exoplanet's mass and radius to explore its relationship with estimated temperatures.

4. Normalization:

Normalization techniques may have been applied if certain algorithms or analyses required data to be on a similar scale. However, based on the provided code, explicit normalization steps were not performed.

5. Insights Gained:

Through exploratory data analysis and visualization, insights were gained into the distribution, variability, and relationships within the dataset.

Temporal trends in planetary detection methods were identified, providing context for the evolution of exoplanet detection techniques.

The relationship between key features such as mass, radius, orbital period, and planetary habitability categories was explored, contributing to our understanding of exoplanetary systems.

The scatter plot comparing estimated temperatures with escape velocities provided insights into the thermal properties and atmospheric retention capabilities of exoplanets.

Question. List any external libraries used in the project

1. **LabelEncoder** from scikit-learn.preprocessing: Used for encoding of categorical columns.

2. **GridSearchCV** from scikit-learn.model_selection: Utilized for hyperparameter tuning through exhaustive search over specified parameter values for an estimator.

3. **KNN** from scikit-learn.ensemble: Employed as the classifier algorithm for the classification task.

4. **SMOTETomek** from imbalanced-learn.over_sampling: Used for Synthetic Minority Over-sampling Technique for balancing class distribution by generating synthetic samples.

5. **PCA** from imbalanced-learn.under_sampling: Principal Component Analysis used for getting important feature column.