Detecting Habitable Exoplanets - A Comparative Study

Harpreet Kour, Naima Noor, Saurav Jayakumar

15th December, 2022

1 Overview

The project aims to present a case study on Data Mining in Astronomy to classify exoplanets in our universe. It is a comparative study to observe the performance and accuracy of various algorithms.

2 Motivation

The observable universe is around 46.5 billion light years across in radius and contains around 100 billion galaxies. Our Milky Way galaxy itself has an estimated 200 to 400 billion stars and around 2 to 3 billion of those stars are similar to our Sun[1]. Given these astronomically large numbers, there ought to be some stars with planets having conditions that are suitable for the evolution of life or at least support life in the form we know. The immenseness of the universe makes it perfect for endless possibilities and that is what excites us the most. This in itself is an exciting reason to be researching exoplanets or being "planet hunters" so to say.

3 Description of the Problem

This project aims to use data available in the NASA exoplanet and planetary systems to predict whether an exoplanet is habitable. This data is raw so we will use data-cleaning techniques and approximations to get it into a usable form. We will initially use planet masses, radius, stellar flux, surface temperature, orbital period, and the distances of each planet from its star to check whether it is in the Goldilocks zone. These parameters can help in determining whether the planet lies inside the Goldilocks zone or not. But we would also be collecting other available data to check whether those parameters could improve the accuracy of the model. Using different models on our cleaned data, we will perform a comparative study for performance and accuracy and present our conclusions in the form of a graph.

The main processes that we are covering as part of this project are as follows:

1. Data cleaning and transformation
2. Analysis and data visualization
3. Applying Principal Component Analysis (PCA) for feature reduction
4. Modeling different classification algorithms
5. Result comparison and reporting

At the end of this project, our aim is to answer questions like 'Is an exoplanet habitable or not?', and 'Which type of exoplanets are usually habitable?'.

4 Goals/Success

This project aims to use data available in the NASA exoplanet archives and planetary systems to predict whether an exoplanet is habitable. This data is raw so we will use data-cleaning techniques and approximations to get it into a usable form. We will then perform PCA to identify correlated features and remove them. Finally, the cleaned data will be used to classify the exoplanets using different models and perform a comparative study for performance and accuracy. With the results from the models, we will present our conclusions in the form of an interactive graph.

5 Data Collection

We are using open-source data sets available for our project. The current data sets that we are using are provided below with a link to the data set(s).

- NASA Exoplanet Archives
- Habitable Exoplanet Catalog

From the 'Habitable exoplanets catalog'[2], we get the list of currently classified habitable planets for predicting the habitability of an exoplanet, and from the 'NASA exoplanet archive', we get additional parameters and details about all known planets. The planets classified as potentially habitable or optimistically potential are assumed to be habitable and the remaining planets are not habitable. This data would then be used for training and testing the model.

All parameters would ideally be constant as values like the mass and radius of a planet would not change. But since the values available in the dataset are observed from a great distance and calculated with many assumptions, there is some probability that the values are incorrect or random. Moreover, if we calculate the mass from the radius or vice-versa, then the calculated value is dependent and could be incorrect.

The masses and radii of exoplanets are fundamental quantities needed for their characterization. Researchers have found that for low masses exoplanets the radius increases with increasing mass, and for large exoplanets, the radius is almost independent of the mass [3]. However, there are some planetary systems that may not follow this linear relationship between mass and radius [4], but we are assuming it is linear. Some planets may not follow the basic mass-radius pattern in the planetary system. However, we are assuming every collection follows the same pattern.

6 Feature Selection

For detecting the habitability of exoplanets, we are using Number of Stars, Number of Planets, Number of Moons, Orbital Period, Planet Radius [Earth Radius], Planet Mass, Stellar Effective Temperature, Stellar Luminosity, and Stellar Surface Gravity as our features. However, the Orbital Period, Planet Radius, and Planet Mass will be the main features for the detection.

As per our preliminary analysis of the data:

- The number of stars is only related to the number of planets and with the rest of the features, it has a negative correlation (and vice versa for the number of planets).
- The number of moons is an independent feature in our data set as it has no relation to any of the other features.
- The orbital period has a strong correlation with stellar effective temperature.
- Planet mass is showing a strong correlation with luminosity and temperature. This means the greater the mass, the hotter and brighter the exoplanet [5].
- Planets having large radii have high luminosity, as shown in figure 1.

	sy_snum	sy_pnum	sy_mnum	pl_orbper	pl_rade	pl_bmasse	st_teff	st_lum	st_logg
sy_snum	1	0.18215	NaN	-0.15112	-0.075584	-0.11932	-0.16356	-0.10718	-0.042209
sy_pnum	0.18215	1	NaN	-0.24024	-0.35149	0.021356	-0.40348	-0.41189	0.34345
sy_mnum	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
pl_orbper	-0.15112	-0.24024	NaN	1	0.32969	0.46497	0.9051	0.83477	-0.76853
pl_rade	-0.075584	-0.35149	NaN	0.32969	1	0.29454	0.4845	0.58206	-0.61632
pl_bmasse	-0.11932	0.021356	NaN	0.46497	0.29454	1	0.4697	0.44176	-0.378
st_teff	-0.16356	-0.40348	NaN	0.9051	0.4845	0.4697	1	0.95678	-0.88314
st_lum	-0.10718	-0.41189	NaN	0.83477	0.58206	0.44176	0.95678	1	-0.95114
st_logg	-0.042209	0.34345	NaN	-0.76853	-0.61632	-0.378	-0.88314	-0.95114	1

Figure 1: Correlation Matrix our data set

7 Preprocessing

Data preparation is of utmost importance before the application of any machine-learning algorithm. Data preprocessing involves the construction and transformation of datasets so that machine-learning algorithms can be applied to understand accurate patterns in the data. Every machine-learning problem focuses on prioritizing the improvement of the quality and size of the dataset.

Our dataset contains features that are missing 40% of its values. Missing values can be handled by replacing them with the mean value of the column data. However, in the detection of exoplanets, this method doesn't give great results for Mass and Radius, which are two main fundamental properties. Since only for a relatively small fraction of exoplanets are they both available [3], we are computing the missing mass and radius value by using the following formulas of the Mass – Radius relation of Chen & Kipping (2017), as shown in figure 2 and 3. The formula uses the planet's mass data for computing radius data and vice versa.

$$\mathcal{R} = \mathcal{C} + \mathcal{M} \times \mathcal{S}, \text{ where}$$

$\mathcal{R} = \log_{10}(R_p/R_\oplus)$, $\mathcal{C}$ = a constant term (in $\log_{10}$ units), $\mathcal{M} = \log_{10}(M_p/M_\oplus)$, $\mathcal{S}$ = the slope of the power-law relation, $R_\oplus$ represents the radius of the Earth, and $M_\oplus$ represents the mass of the Earth. We use the following parameters, which are provided in their Table 2 (or derived thereof), to compute a planet radius R_p given an input, empirically determined planet mass M_p:

$$\{\mathcal{C};\ \mathcal{S}\} = \begin{cases} \{0.00346;\ \ 0.2790\} & \text{if } \ M_p < 2.04 M_\oplus \\ \{-0.0925;\ \ 0.589\} & \text{if } \ 2.04 \leq M_p/M_\oplus < 132 \\ \{1.25;\ \ -0.044\} & \text{if } \ 132 \leq M_p/M_\oplus < 26600 \\ \{-2.85;\ \ 0.881\} & \text{if } \ M_p \geq 26600 M_\oplus. \end{cases}$$

Figure 2: Formula for calculating Radius using Mass

$$\mathcal{M} = (\mathcal{R} - \mathcal{C})/\mathcal{S}.$$

However, it is crucial to note that, since the third term of the mass-radius relation has a negative slope (i.e., $\mathcal{S} < 0$), there exists a regime of planet radius R_p for which a given planet radius R_p does not uniquely map into a single planet mass M_p. This inherently degenerate regime spans approximately $11.1 \leq R_p/R_\oplus \leq 14.3$. We thus abstain from calculating a planet mass M_p in this regime, which spans multiple orders of magnitude (in planet mass M_p). The resulting piecewise function is thus:

$$\{\mathcal{C};\ \mathcal{S}\} = \begin{cases} \{0.00346;\ \ 0.2790\} & \text{if } \ R_p < 1.23 R_\oplus \\ \{-0.0925;\ \ 0.589\} & \text{if } \ 1.23 \leq R_p/R_\oplus < 11.1 \\ \{-2.85;\ \ 0.881\} & \text{if } \ R_p \geq 14.3 R_\oplus. \end{cases}$$

Figure 3: Formula for calculating Mass using Radius

After cleaning and preprocessing our dataset, we have 4970 data points. 57 are the observations of habitable planets, and 4913 belong to non-habitable planets.

7.1 Plotting of Data

7.1.1 2D Scatter Plot

We have used 2D scatter plots to analyze the relationship between features. For example, it is obvious from the graphs that the feature: "sy_mnum" does not contribute to deciding whether an exoplanet is habitable as it is a constant value for all planets. Similarly, these plots will allow us to understand the correlations between the various features and help in selecting features that can be removed from the dataset to improve the performance, as shown in Fig 4.

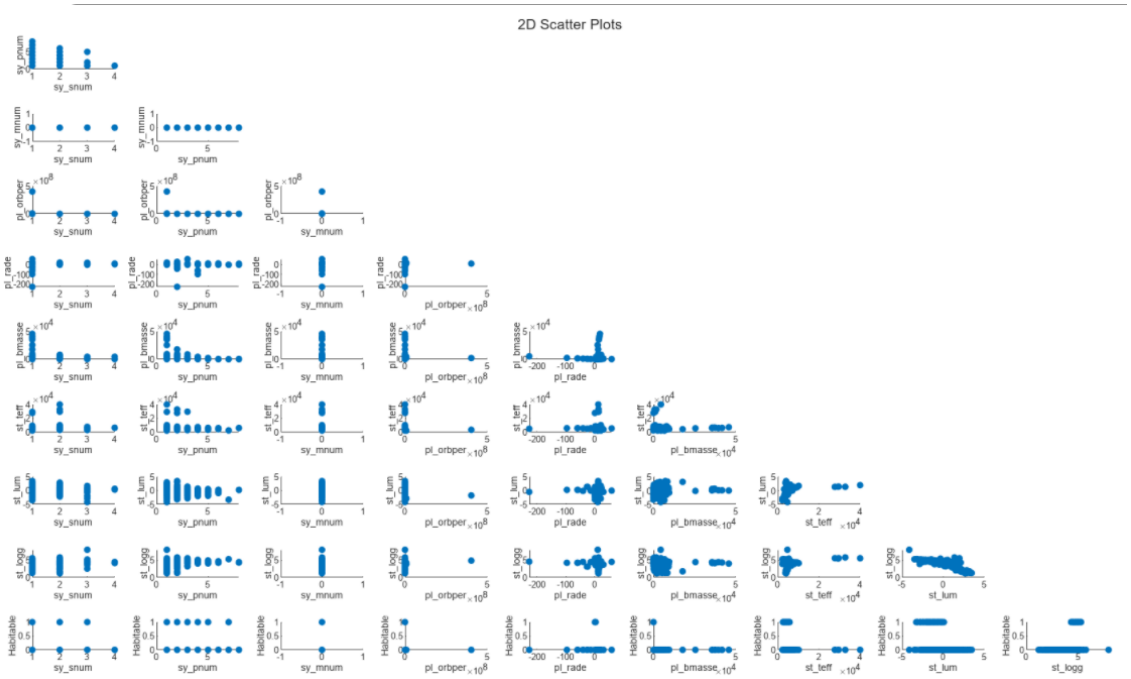

Figure 4: 2D Scatter Plot in the original space

7.1.2 3D Scatter Plot

We have used the 3D scatter plots to check if our data can be separated based on any particular features. 3D plots are also plotted in the transformed spaces - Ur and U post PCA. This will help us to see if any transformed features might be able to classify our data better than the original features. First, we plotted the graph on our entire dataset. Since we have an imbalanced dataset with habitable planets being very less compared to the non-habitable ones, we have plotted the same graph with an equal number of observations from both habitable and non-habitable classes. Fig 5, 6

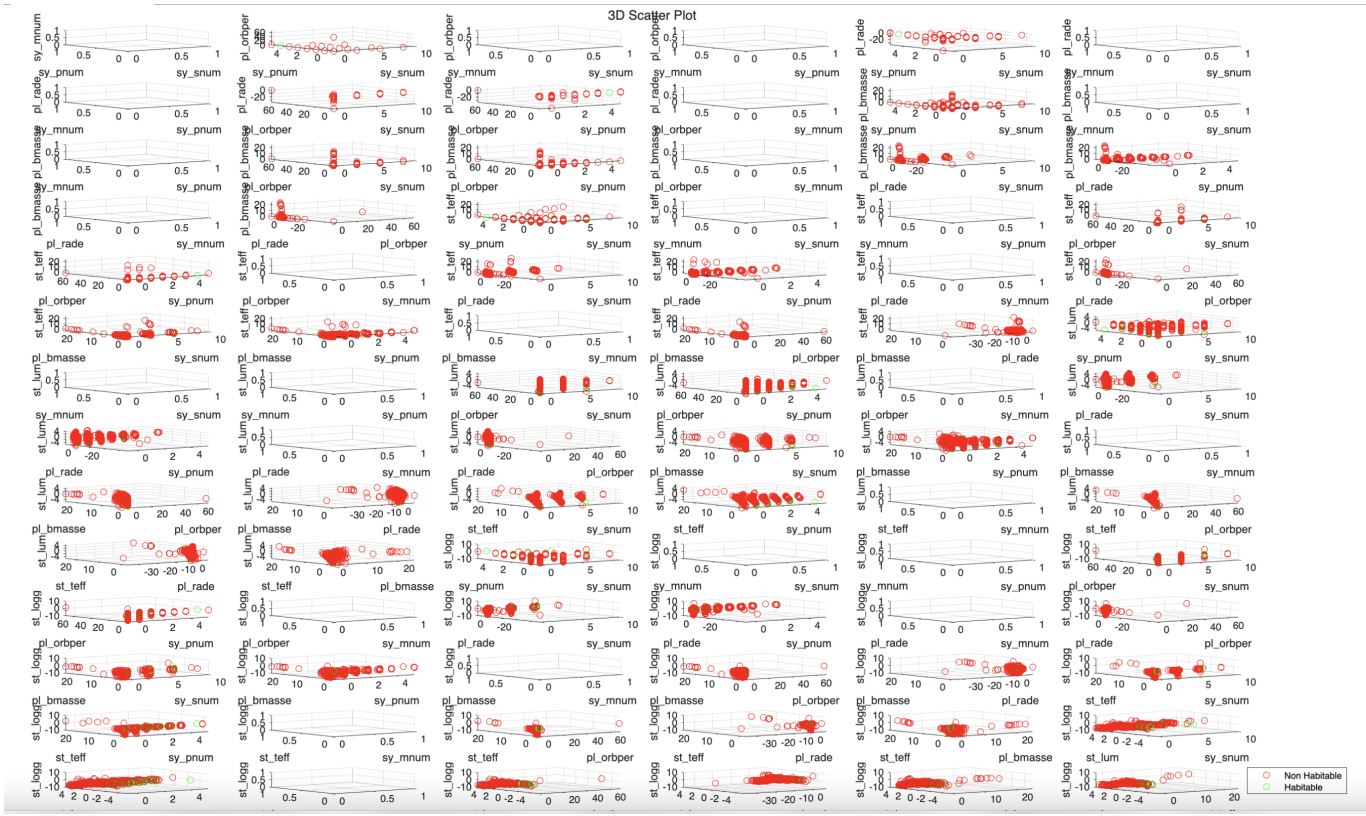

Figure 5: 3D Scatter Plot in the original space

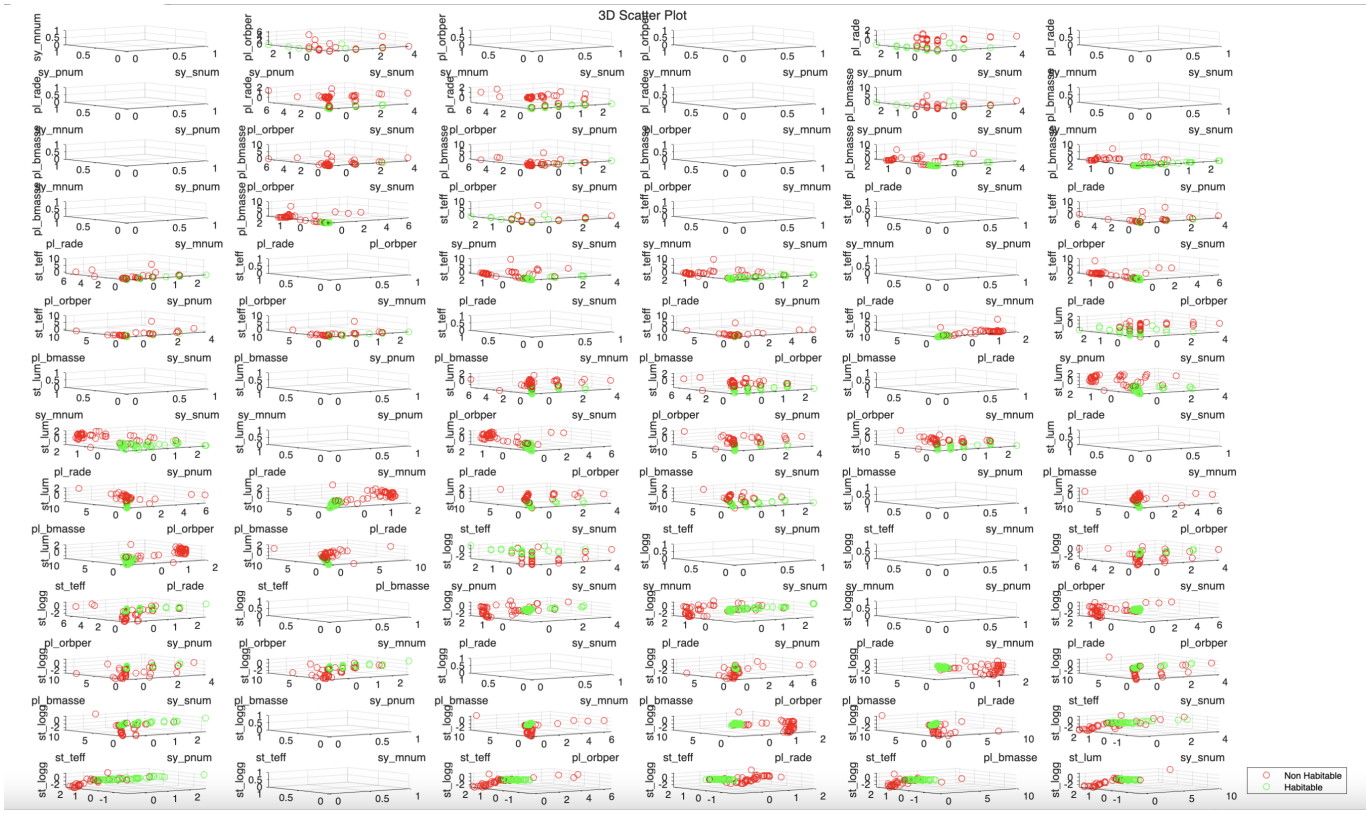

Figure 6: 3D Scatter Plot in the original space with equal observations from both classes

7.1.3 Normal Probability Density Function Graphs

These graphs are shown in Fig 7. It is also combined with a 1D plot of the data points of each feature. This shows us the normal distribution of our data and also the irregularities if any.

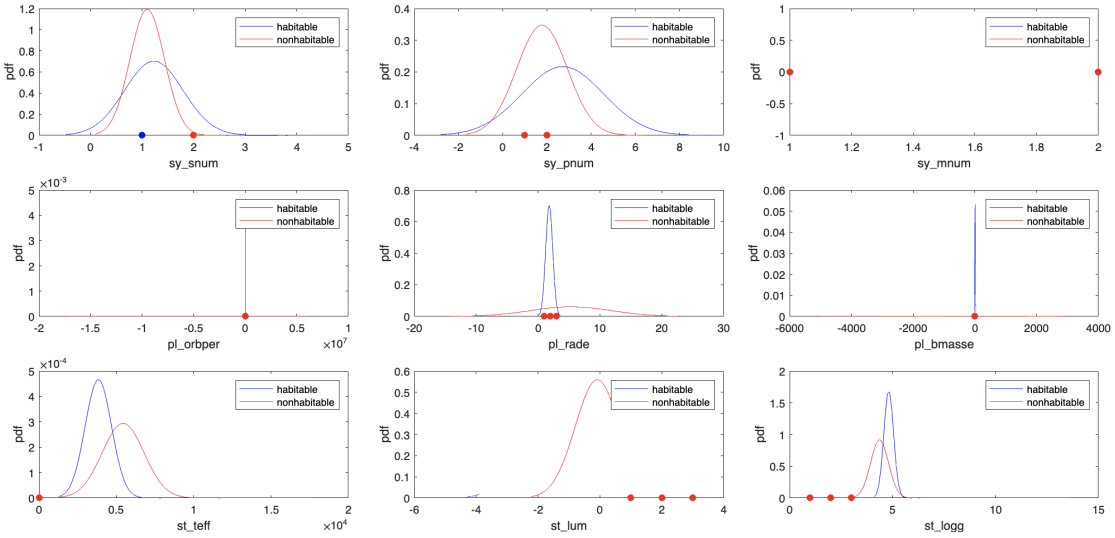

Figure 7: Normal PDF Plots

7.2 Analysis of the Plots

7.2.1 2D Scatter Plots

From the 2D scatter plots we can see that all the graphs related to the number of suns and planets in the system have distinct values and there is no trend that can be spotted. The graphs involving the number of moons show a constant in the corresponding axis as the values of this column in all the observations are 0. Some sort of trend is visible in the graphs with radius, mass, and stellar features. Among them, we can see that stellar features have formed a clear divide when compared to the habitable column. Luminosity and Surface temperature of the star are positively correlated for the initial values. The rest of the values can be outliers as they are not following any pattern. If we look at stellar features vs mass of the planet we can see that majority of our data is clustered around the lower mass region. This suggests that the planet's lower mass is correlated with its star's luminosity and surface gravity.

7.2.2 3D Scatter Plots

From the original 3D scatter plots, we cannot really see a separation between the two classes. All the graphs that contain the sy_mnum feature have no points because all the data is zero in this feature. This feature can be removed. Other than that in the original graph with all the data we can hardly see the habitable class. Hence, we are using the graph with similar observations for further analysis. All the graphs in the last row can be separated into two clusters if viewed from a different angle. The common feature in them is st_logg - stellar surface gravity. In another graph (1st row 5th column) that is separable over height, the features involved are sy_pnum, sy_snum, pl_rade. This tells us that these features will be important in determining habitability. Also by looking at the graphs from the transformed spaces it can be said that the original features are better at distinction when compared to the transformed features.

7.2.3 Normal Probability Density Function Graphs

From these graphs, we can see the normal distribution of our data. We can see that the number of planets and number of stars in the system are almost equivalent in both classes but the variability of the habitable class is more. The number of moons graph does not show any curve because all of the data is the same which is 0. This column can be removed. From the orbital period, mass, and

radius graphs we can see a high standard deviation in the non-habitable class whereas the habitable class is mostly clustered around its mean. The last three graphs deal with the stellar information of the system around which the exoplanet is revolving namely, stellar temperature, luminosity, and surface gravity. For the habitable class, the data is mostly clustered around the mean. This tells us that only a few types of suns support the conditions for a planet to be potentially habitable. We have also plotted the data points but most of them overlap in both classes or are extremely small values hence not represented on the graphs in a way that we can see them. For the last three graphs, the visible data points lie outside the curves which is not the case for the initial six graphs.

7.3 PCA

In order to reduce the number of dimensions in our original feature space, we have used PCA on the dataset and generated the below plots:

7.3.1 Scree Plots

From the scree plots, we can observe that the 2nd to 6th Principal Components provide equal amounts of information and with the first 6 Principal Components, we retain 90% of the information. Hence, if we are using the dataset after PCA transformation, we will be using only these 6 features in the new space for our model.

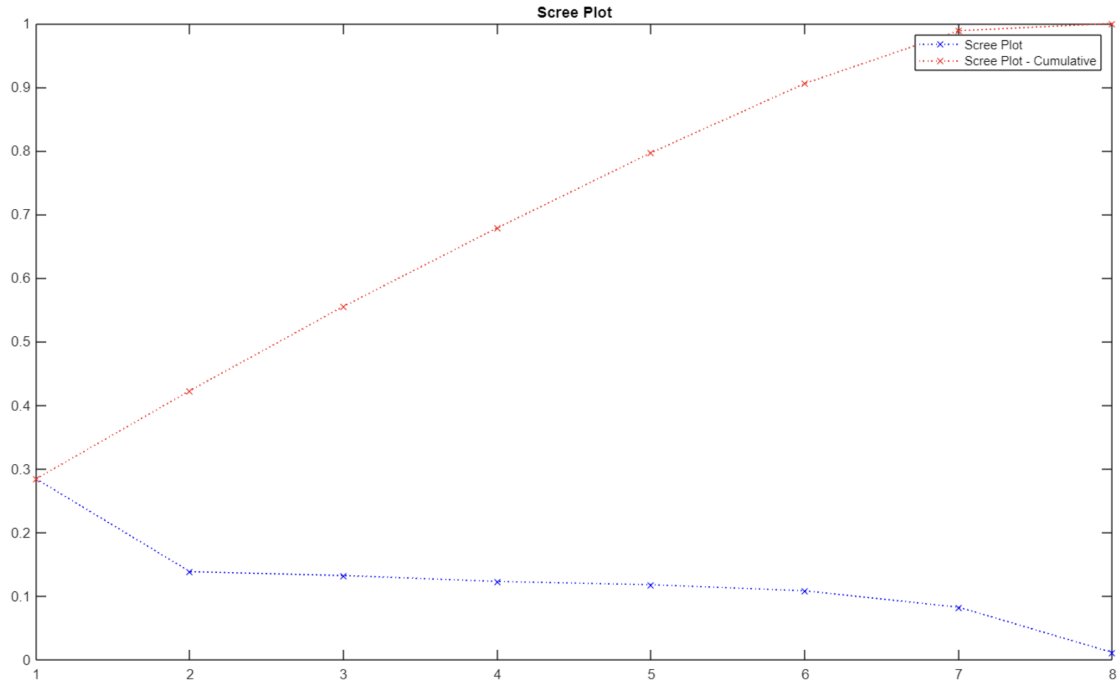

Figure 8: Scree Plots

7.3.2 3D Scatter Plots

3D scatter plots in the transformed spaces - Ur and U can be seen below. As is clear from the scree plots we only need till PC6 to get 90% of our data. Therefore we have only plotted the graphs in the transformed space up to PC6. We have used the same number of observations from both classes in order to balance our data. Doing this has helped us visualize the **habitable** class better. We can see from the graphs below that both classes are overlapping. Even the transformed space is not giving us a clear distinction between the two classes. We can also spot some outliers from both classes in the transformed spaces.

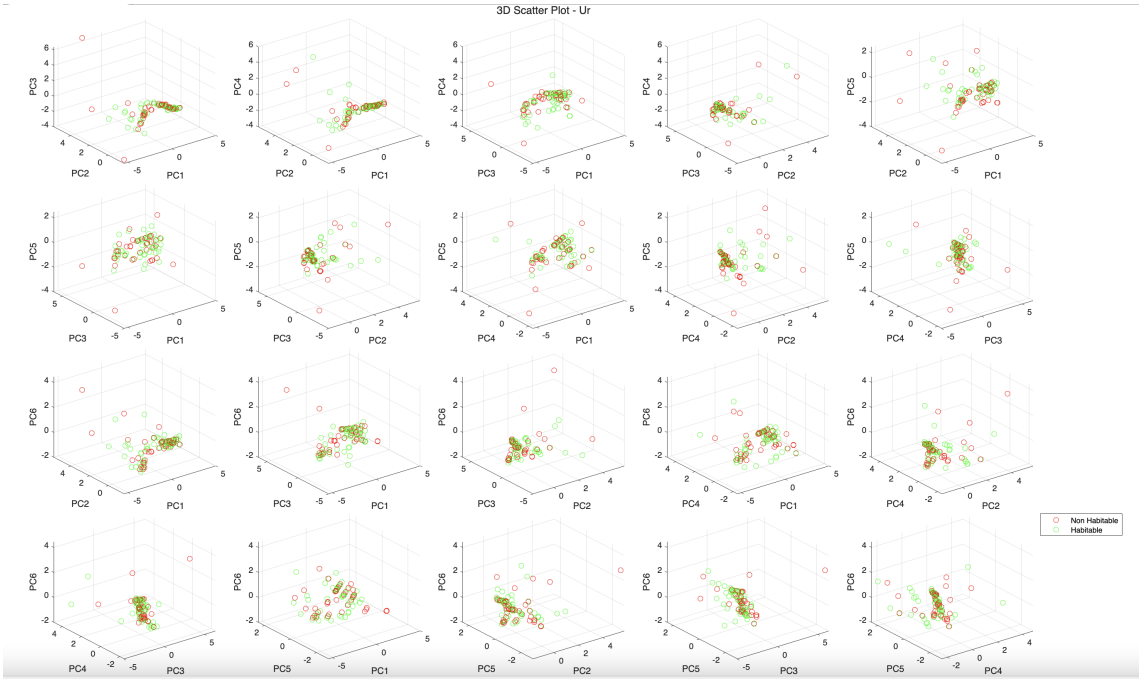

Figure 9: Transformed Space - Ur

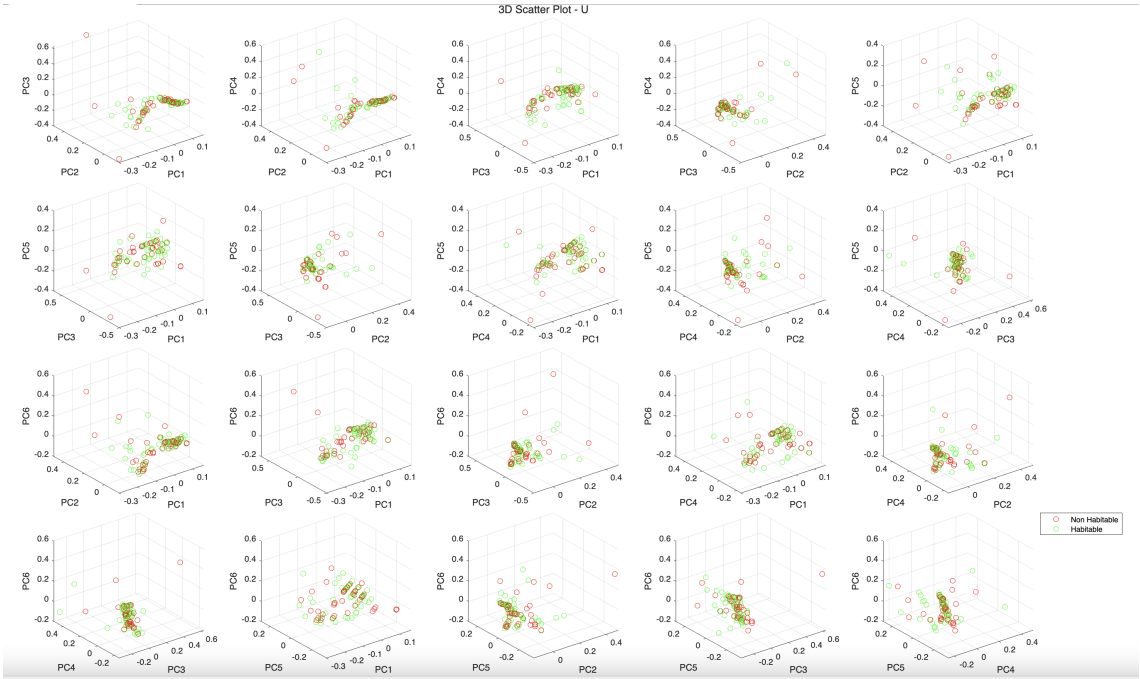

Figure 10: Transformed Space - U

7.3.3 Loading Vectors

When we say positively correlated it can mean either in the positive direction or negative direction, inversely correlated means an increase in one feature causes a decrease in the other.

Below is what each feature in the loading vectors represents.

Feature	Name
Feature 1	Number of Stars
Feature 2	Number of Planets
Feature 3	Orbital Period
Feature 4	Planet Radius
Feature 5	Planet Mass
Feature 6	Stellar Effective Temperature
Feature 7	Stellar Luminosity
Feature 8	Stellar Surface Gravity

Analysis of each loading vector plot

- **Loading Vector 1**: We can see that the Number of Stars, Radius, Mass, Temperature, and Luminosity positively correlate. Luminosity has the most significant impact on this loading vector. Luminosity and surface gravity are inversely correlated to each other with almost the same magnitude. On the other hand, Orbital Period has almost no contribution to this loading vector, as shown in figure 11.

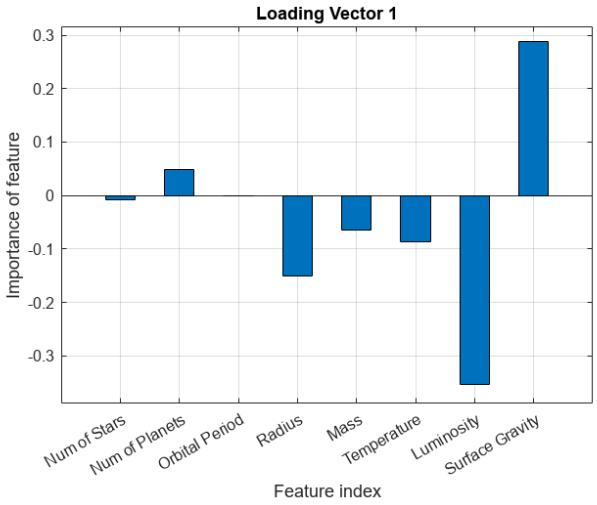

Figure 11: Loading Vector 1

- **Loading Vector 2**: Number of Stars has the maximum and Surface Gravity has the minimum contribution on this loading vector. From figure 12, we can see that Number of Stars, Orbital Period, Radius, and Mass are directly related to each other and so do Number of Planets, Temperature, and Luminosity. Luminosity and Surface Gravity are again inversely correlated to each other. However, Luminosity has the greatest magnitude than Surface Gravity. Mass and Temperature also show an inverse correlation.

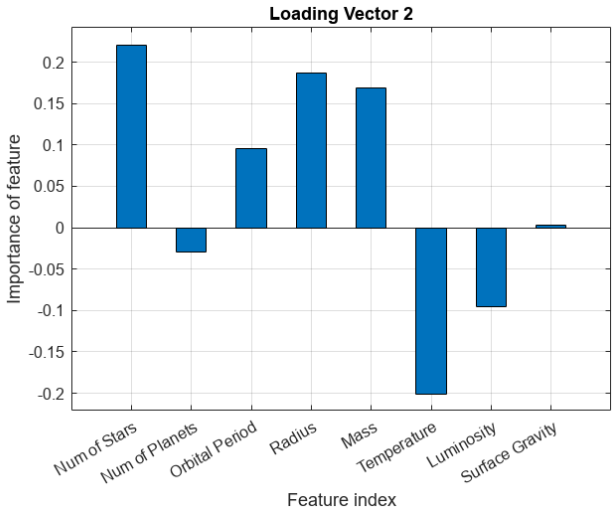

Figure 12: Loading Vector 2

- **Loading Vector 3**: We can see that Number of Stars again has the maximum impact. However, this time it is in the negative direction. Features Number of Stars, Number of Planets, and Temperature show a positive correlation. Radius has the minimum contribution in this loading vector. Mass and Temperature are again inversely related to each other. Number of Stars and Radius are again directly correlated to each other, as shown in figure 13.

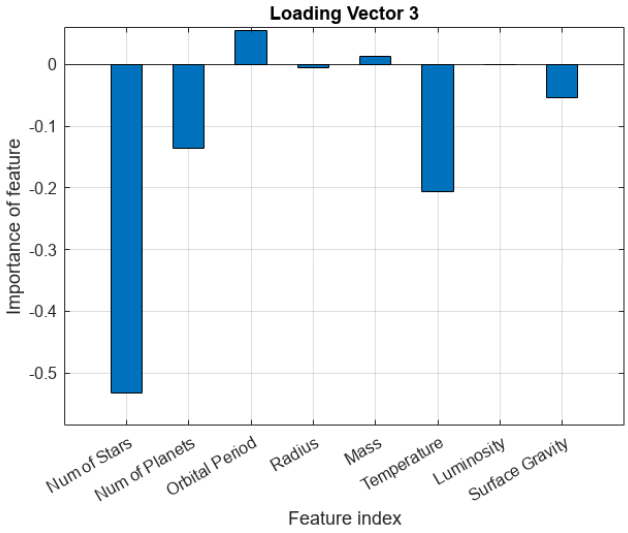

Figure 13: Loading Vector 3

- **Loading Vector 4**: We can see that Orbital Period has the maximum impact. Number of Stars and Number of Planets have zero impact on this loading vector. Other than Orbital Period and Temperature, all other features have very little contribution to this loading vector. Figure 14 illustrates features Mass and Temperature again show the inverse relation. Luminosity and Surface Gravity show positive correlation in this loading vector.

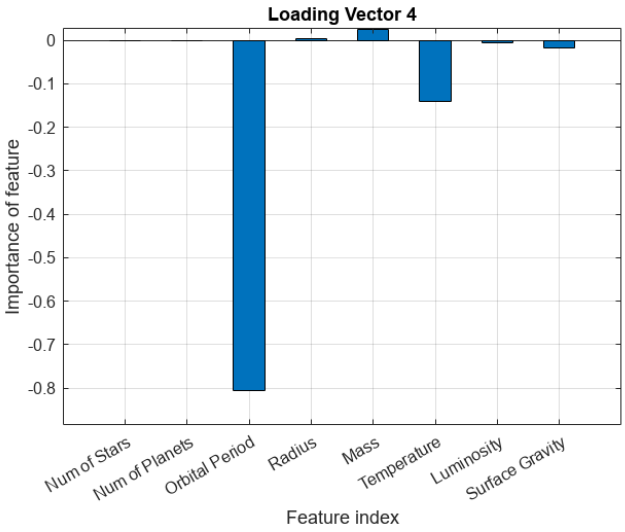

Figure 14: Loading Vector 4

- **Loading Vector 5**: We can see that Number of Planets has the maximum contribution in this loading vector. Number of Stars again has the least contribution in this loading vector. Number of Planets, Orbital Period, Mass, and Luminosity show a positive correlation. Like the other loading vectors, Luminosity and Surface Gravity are again inversely correlated to each other. In this loading vector, Surface Gravity's magnitude is greater than Luminosity Mass and Temperature are again inversely related to each other, as shown in figure 15.

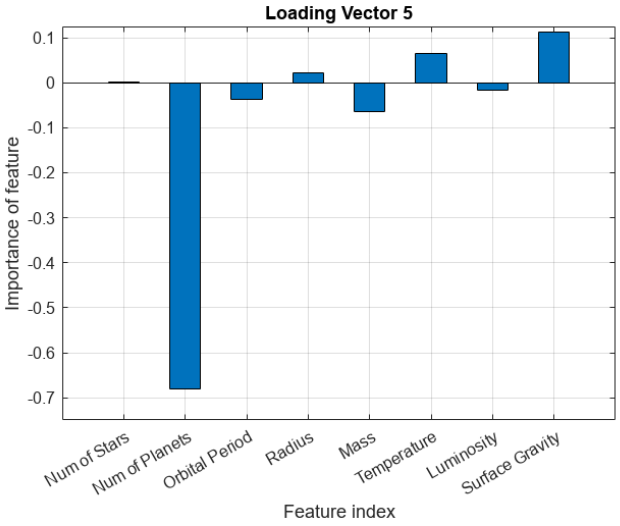

Figure 15: Loading Vector 5

- **Loading Vector 6**: We can see that Mass has the maximum impact. As figure 16 indicates, Number of Planets has the minimum value of loading. Orbital Period and Radius also have less contribution in this loading vector. In this loading vector, Mass and Temperature are directly related to each other. Number of Stars and Radius are directly correlated with each other. Luminosity and Surface Gravity are again inversely correlated, with the same magnitude that they have in loading vector 5.

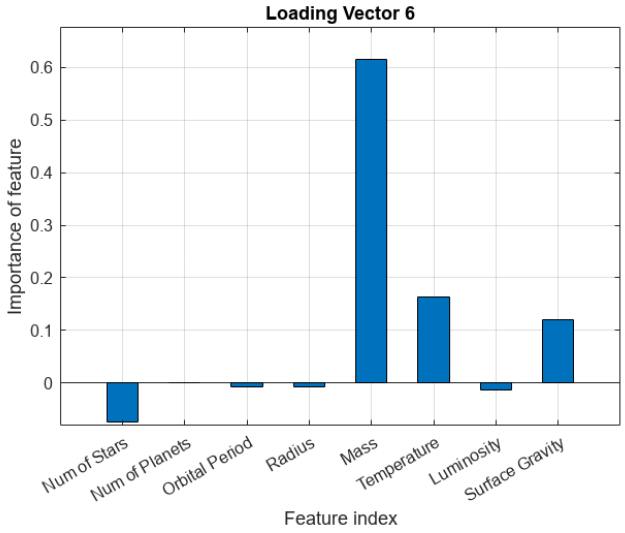

Figure 16: Loading Vector 6

- **Loading Vector 7**: We can see that Radius has the maximum impact with Number of Stars having the second most impact, albeit negatively. As figure 17 indicates, Orbital Period has the minimum value of loading, and Temperature and Luminosity do not contribute much. In this loading vector, Number of Planets and Radius correlate with each other and so do Number of stars and Mass. Luminosity and Surface Gravity again show an inverse correlation, with Surface Gravity having greater magnitude than Luminosity.

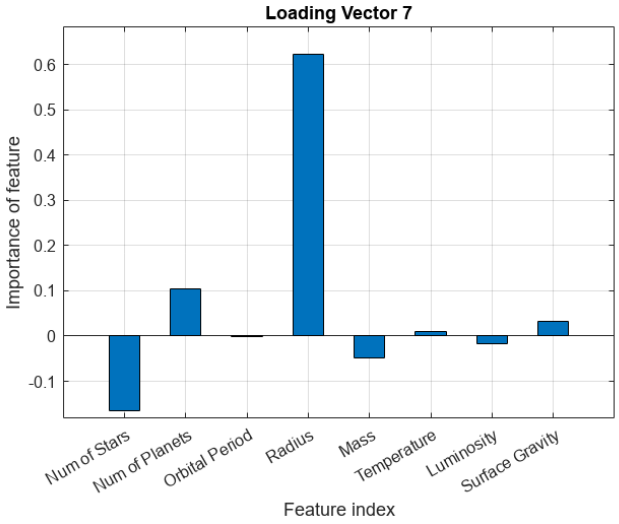

Figure 17: Loading Vector 7

- **Loading Vector 8**: Based on figure 18, we can observe that Number of Stars, Number of Planets, Orbital Period, Radius, and Mass do not contribute at all. Luminosity and Surface Gravity correlates and contributed negatively to this loading vector while Temperature contributes positively.

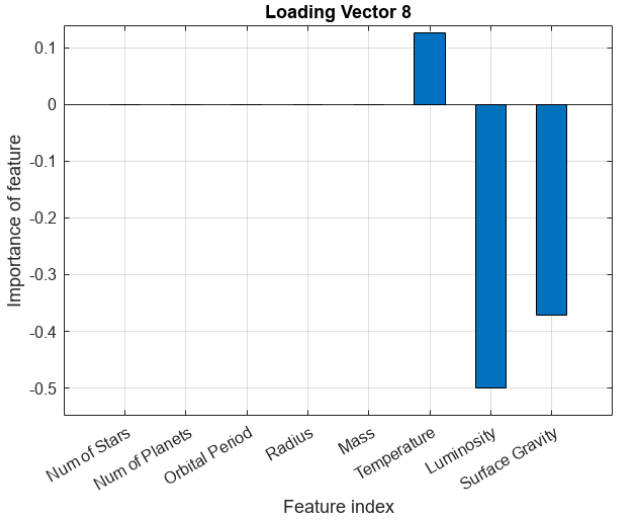

Figure 18: Loading Vector 8

Inferences from the loading vectors:

The process that we followed to get correlated features is as follows: We compared each feature against all other features across all the loading vectors. So, we had 15 observations in total (1,2), (1,3), (1,4) ... till (7,8) (8,8). After checking we came to the following conclusion:

- Number of Stars and Radius are highly correlated to each other. If we observe their pattern across the loading vectors, we can see they are positively correlated in loading vector 1, with Number of Stars having a smaller magnitude than Radius. They are also positively correlated in loading vector 2 but in the opposite direction to loading vector 1. The magnitude of Number of Stars is greater than Radius. Coming to loading vector 3, here Number of Stars again has a greater magnitude than Radius but now they are in opposite directions. In loading vector 5, they are again directly correlated to each other, but Number of Stars has a smaller magnitude than Radius. Similarly in loading vector 6, they again have a direct correlation. After observing the pattern between them, we can say that Planet Radius can be dependent on how many stars it has. As they show a clear pattern in our loading vectors, we can remove the number of stars and still get accurate results.

- Luminosity and Surface Gravity are inversely correlated with each other. If we observe their pattern across the loading vectors, we can see they are inversely correlated in loading vector 1 with almost the same magnitude. There is again an inverse correlation between Luminosity and Surface Gravity in loading vector 2, with Luminosity being larger than Surface Gravity. As can be seen, Luminosity and Surface Gravity appear to be positively correlated in loading vector 4, with Surface Gravity having a larger magnitude. As we move towards loading vector 5, Luminosity and Surface Gravity are again inversely correlated. The magnitude of Surface Gravity is greater than that of Luminosity again. A similar inverse correlation exists between Luminosity and Surface Gravity in loading vector 7, and Surface Gravity is again larger than Luminosity. As they are showing a clear pattern in our loading vectors, we have decided to remove the Luminosity because Surface Gravity has the greater magnitude in most of the loading vectors. As they are showing a pattern so after deleting Luminosity we can still get accurate results.

After looking at our PCA results, we are going forward with the following features: **Number of Planets**, **Orbital Period**, **Planet Radius**, **Planet Mass**, **Stellar Effective Temperature**, and **Stellar Surface Gravity**.

8 Classification

For the classification task, we decided to go ahead with four classifiers. We chose one of the classifiers we discussed in class - **K Nearest Neighbour(KNN)** [6] and the other three which suited best for our particular problem. The other three classifiers we have used are - **Logistic Regression** [7], **Multi-Layer Perceptron Classifier (MLP)** [8], and **Support Vector Machines (SVM)** [9]. Since our data is highly imbalanced (there are very few potentially habitable planets) we have used sampling techniques to resolve the class imbalance. In sampling, either data from the class which has larger values is removed - under-sampling or additional samples are added to the class with smaller values - oversampling. We have used different sampling techniques in our project to achieve the best results. The sampling techniques used are SMOTE [10], ADASYN [11], SMOTE-Tomek [12], and ClusterCentroids [13]. Because we have used multiple classifiers with multiple sampling techniques and different data sets we have 96 rows of results with F1 score, precision, recall, AUC, and confusion matrices, all of which are attached along with our code and report in an excel file. Here we are summarizing our results based on the six different data sets we used. The data sets include all the original features, the features we selected, features in transformed space - Ur including all PCs, features in transformed space - Ur including top 6 PCs, features in transformed space - U including all PCs and features in transformed space - U including top 6 PCs. We had different performance metrics but we are talking about the recall values here as in our case that is the most important metric. We want to know the correctly classified potentially habitable exoplanets. We have used python for all the code work of the classification part of the project.

8.1 PHL - All Features

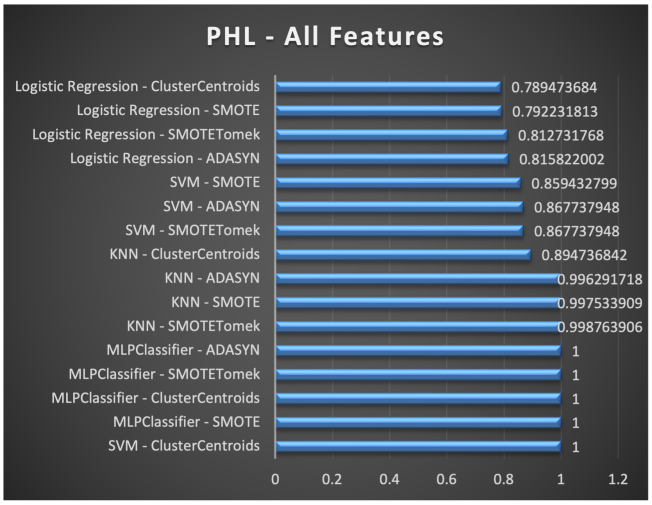

Figure 19: PHL - All Features (Recall)

The best recall values are achieved with the combination of MLP Classifier with all four sampling techniques which is equal to 1. Even SVM with ClusterCentroids perform equally well and have a recall value of 1. Logistic Regression seems to perform the worst among all the algorithms giving a recall value of 0.79. Also, this data set is quite useful in predicting the habitability of the exoplanets.

8.2 PHL - Selected Features

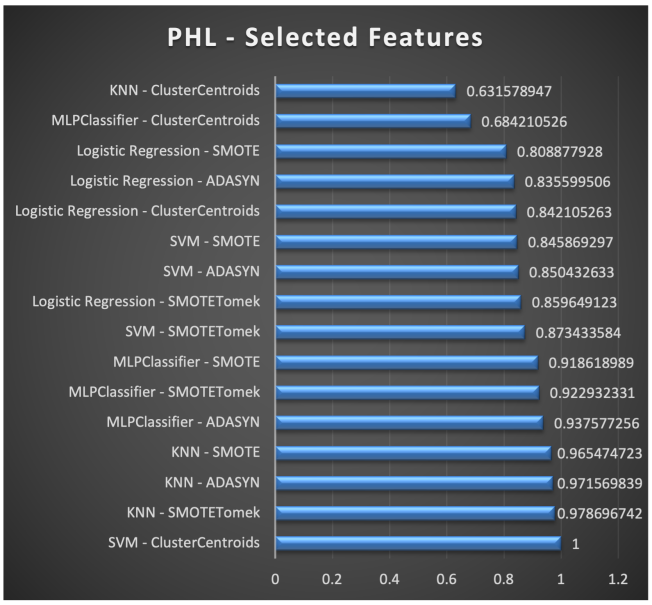

Figure 20: PHL - Selected Features (Recall)

The best recall value is achieved with the combination of SVM classifier using the ClusterCentroids sampling technique which is 1. Even the KNN algorithm performs well and has recall values around 0.97 which is great. MLP classifier and Logistic Regression seem to perform the worst among all the algorithms giving a recall value of around 0.6 and 0.8. This is the dataset in which we also see a difference in using different sampling techniques for example KNN with SMOTETomek gives a very good recall value of 0.97 whereas the same KNN algorithm with ClusterCentroids as the sampling technique gives us a low value of 0.63.

8.3 Ur - All PCs

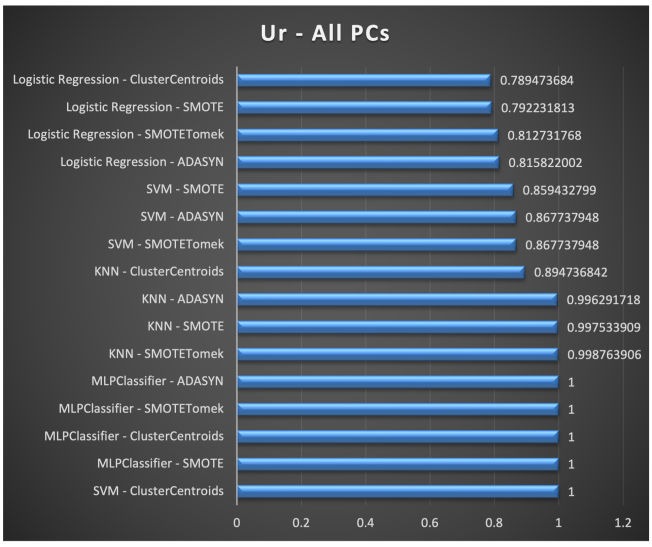

Figure 21: Ur - All PCs (Recall)

In the transformed space - Ur with all the PCs many models are predicting the results accurately. MLP and SVM with ClusterCentroids have a perfect recall value of 1 whereas logistic regression is performing the worst with around 0.78 for the recall value. Values for KNN are also very good at around 0.99.

8.4 Ur - Top 6 PCs

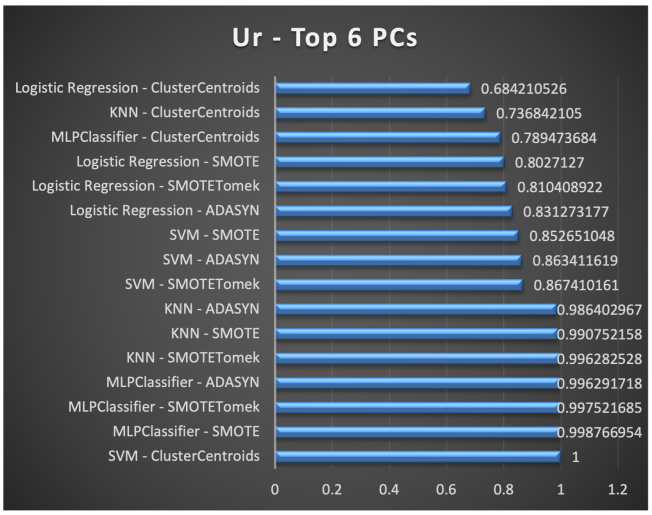

Figure 22: Ur - Top 6 PCs (Recall)

The dataset doesn't seem to provide as much information as the other datasets as only the SVM model trained on an under-sampled original dataset has a perfect recall. But there are other models like the MLPClassifier trained on SMOTE sampled data or SMOTETomek sampled data or ADASYN sampled data which achieve near-perfect recall. This means that we can get the required information from the Top 6 PCs using the Ur matrix but only when using certain models trained on certain types of sampled data.

8.5 U - All PCs

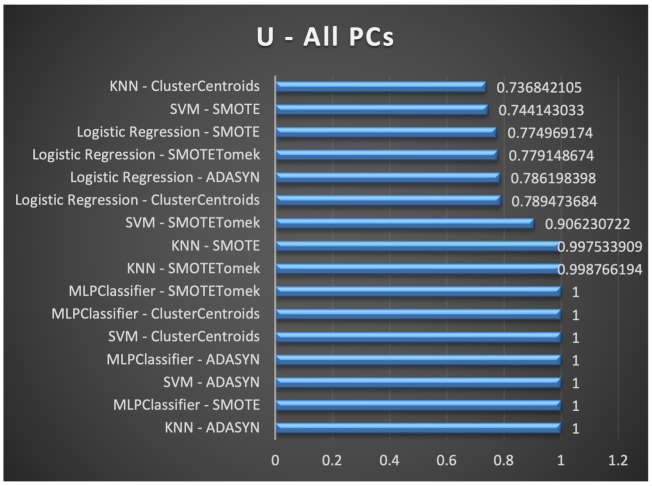

Figure 23: U - All PCs (Recall)

The U matrix with all PCs contains enough information to allow many models to predict the positive class perfectly. We can see that KNN, SVM, and MLPClassifier models give the best performance when trained on SMOTE, ADASYN, or Clustercentroid sampled datasets. Even the other model-sampler combinations perform decently with the least getting more than 70% recall.

8.6 U - Top 6 PCs

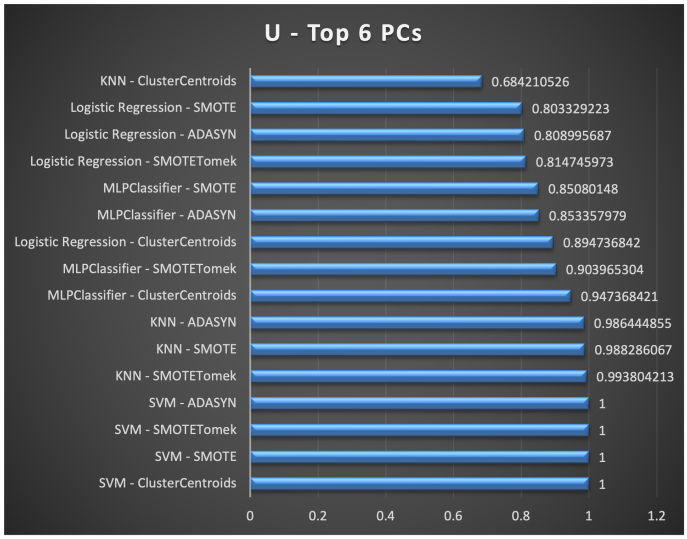

Figure 24: U - Top 6 PCs (Recall)

When using the U matrix with the top 6 PCs, only the SVM model gives the best recall for all 4 sampling methods. The second-best performance is given by the KNN model. Even though the SVM model gets a perfect recall, Logistic regression models and ClusterCentroid-trained KNN model struggle to get good recall values, with the KNN model getting the worst recall score of 68%.

9 Conclusion

Comparing the datasets, models, and samplers [14], we can see that the 'U - All PCs', 'PHL - All Features', and 'Ur - All PCs' datasets provide the most information, the KNN model consistently get high F1 and recall scores, and the ADASYN, SMOTE, and SMOTETomek resampling methods generate the best distribution for the models. In contrast, the Logistic Regression model performs the worst with 80% recall compared to all datasets and samplers providing good recall scores throughout. The best model in our test set turns out to be the KNN model trained on ADASYN sampled U - All PCs dataset with 0.98 F1-score, 96% precision, 100% recall, and 0.98 AUC. The model also trains very quickly as it only takes 0.28 seconds for the entire run. This is the model that we propose as the solution to predicting the habitability of an exoplanet given its features in the U matrix format.

Dataset	Model	Sampler	F1 Score	Precision	Sensitivity/Recall(TPR)	Specificity(TNR)	AUC	Time(sec)
U - All PCs	KNN	ADASYN	0.981257557	0.963204748	1	1	0.980887793	0.275402069
PHL - All Features	SVM	ClusterCentroids	0.716981132	0.558823529	1	1	0.605263158	1.122105837
Ur - All PCs	SVM	ClusterCentroids	0.716981132	0.558823529	1	1	0.605263158	1.341802835
U - Top 6 PCs	SVM	ClusterCentroids	0.690909091	0.527777778	1	1	0.552631579	2.265093088
Ur - Top 6 PCs	SVM	ClusterCentroids	0.690909091	0.527777778	1	1	0.552631579	2.732113838

Figure 25: Top 5 Dataset - Model - Sampler

10 Team Assessment

10.1 Iteration 1

- Harpreet Kour - I worked on the ideation part of the project. The detailed processes that we will be studying as part of this and the questions that we hope to answer by the end of this project. I also found the data sets for Kepler and TESS projects that will be used for showcasing the transit method of exoplanet detection.
- Naima Noor - I looked through the fundamental information to gather materials in regards to all the conceivable to control the given information as per our prerequisite. I also worked on noise filtration and documentation of the project.

- Saurav Jayakumar - I analyzed the datasets to determine the important parameters for determining whether an exoplanet is habitable or not and combined the PHL dataset with the NASA Exoplanet dataset for additional parameters. I have also developed a plan to clean the data and fill in the missing values.

10.2 Iteration 2

- Harpreet Kour - I worked majorly on the analysis of the plots along with drawing up the plots for normal distribution.
- Naima Noor - I worked on the documentation of the project and along with that I analyzed our features and determine the relationships between them.
- Saurav Jayakumar - I worked on the clean-up and analysis of the raw data, determining the appropriate plots, and plotting them for further analysis.

10.3 Iteration 3

- Harpreet Kour - I worked on the coding, plotting, and analysis for the 3D scatter plots in both original space and transformed spaces.
- Naima Noor - I was responsible for the coding and analysis of loading vectors of PCA.
- Saurav Jayakumar - I worked on the code for data preprocessing and PCA along with the plotting and analysis of Scree Plots. I also merged the 3 codes for the final submission.

10.4 Iteration 4

- Harpreet Kour - I worked on the KNN algorithm and its results. I was also responsible for the representation of the performance metrics in tableau.
- Naima Noor - I worked on the Logistic Regression algorithm as well as its evaluation. Along with that the slides for the final presentation.
- Saurav Jayakumar - I worked on building the resampling and dataset selection architecture for testing the different models and built the SVM and MLPClassifier models. Also, I prepared the performance result report and the associated plots and wrote the Conclusion for the report.

11 Appendix

Goldilocks zone: habitable zone, is the range of distance with the right temperatures for water to remain liquid.
Stellar: relating to a star or stars.

References

[1] Geoffrey W Marcy, Lauren M Weiss, Erik A Petigura, Howard Isaacson, Andrew W Howard, and Lars A Buchhave. Occurrence and core-envelope structure of 1–4× earth-size planets around sun-like stars. *Proceedings of the National Academy of Sciences*, 111(35):12655–12660, 2014.

[2] Phl @ upr arecibo - the habitable exoplanets catalog. https://phl.upr.edu/projects/habitable-exoplanets-catalog. (Accessed on 11/08/2022).

[3] Dolev Bashi, Ravit Helled, Shay Zucker, and Christoph Mordasini. Two empirical regimes of the planetary mass-radius relation. *Astronomy & Astrophysics*, 604:A83, 2017.

[4] Damian C Swift, JH Eggert, Damien G Hicks, Sebastien Hamel, Kyle Caspersen, Eric Schwegler, Gilbert W Collins, Nadine Nettelmann, and GJ Ackland. Mass–radius relationships for exoplanets. *The Astrophysical Journal*, 744(1):59, 2011.

[5] Ines Brott, Selma E de Mink, Matteo Cantiello, Norbert Langer, Alex de Koter, Chris J Evans, Ian Hunter, Carrie Trundle, and Jorick S Vink. Rotating massive main-sequence stars-i. grids of evolutionary models and isochrones. *Astronomy & Astrophysics*, 530:A115, 2011.

[6] sklearn.neighbors.kneighborsclassifier — scikit-learn 1.2.0 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. (Accessed on 12/15/2022).

[7] sklearn.linear_model.logisticregression — scikit-learn 1.2.0 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. (Accessed on 12/15/2022).

[8] sklearn.neural_network.mlpclassifier — scikit-learn 1.2.0 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. (Accessed on 12/15/2022).

[9] sklearn.svm.svc — scikit-learn 1.2.0 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. (Accessed on 12/15/2022).

[10] Smote — version 0.10.0. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html. (Accessed on 12/15/2022).

[11] Adasyn — version 0.11.0.dev0. https://imbalanced-learn.org/dev/references/generated/imblearn.over_sampling.ADASYN.html. (Accessed on 12/15/2022).

[12] Smotetomek — version 0.11.0.dev0. https://imbalanced-learn.org/dev/references/generated/imblearn.combine.SMOTETomek.html. (Accessed on 12/15/2022).

[13] Clustercentroids — version 0.10.0. https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.ClusterCentroids.html. (Accessed on 12/15/2022).

[14] Free data visualization software | tableau public. https://public.tableau.com/app/resources/community-resources. (Accessed on 12/15/2022).