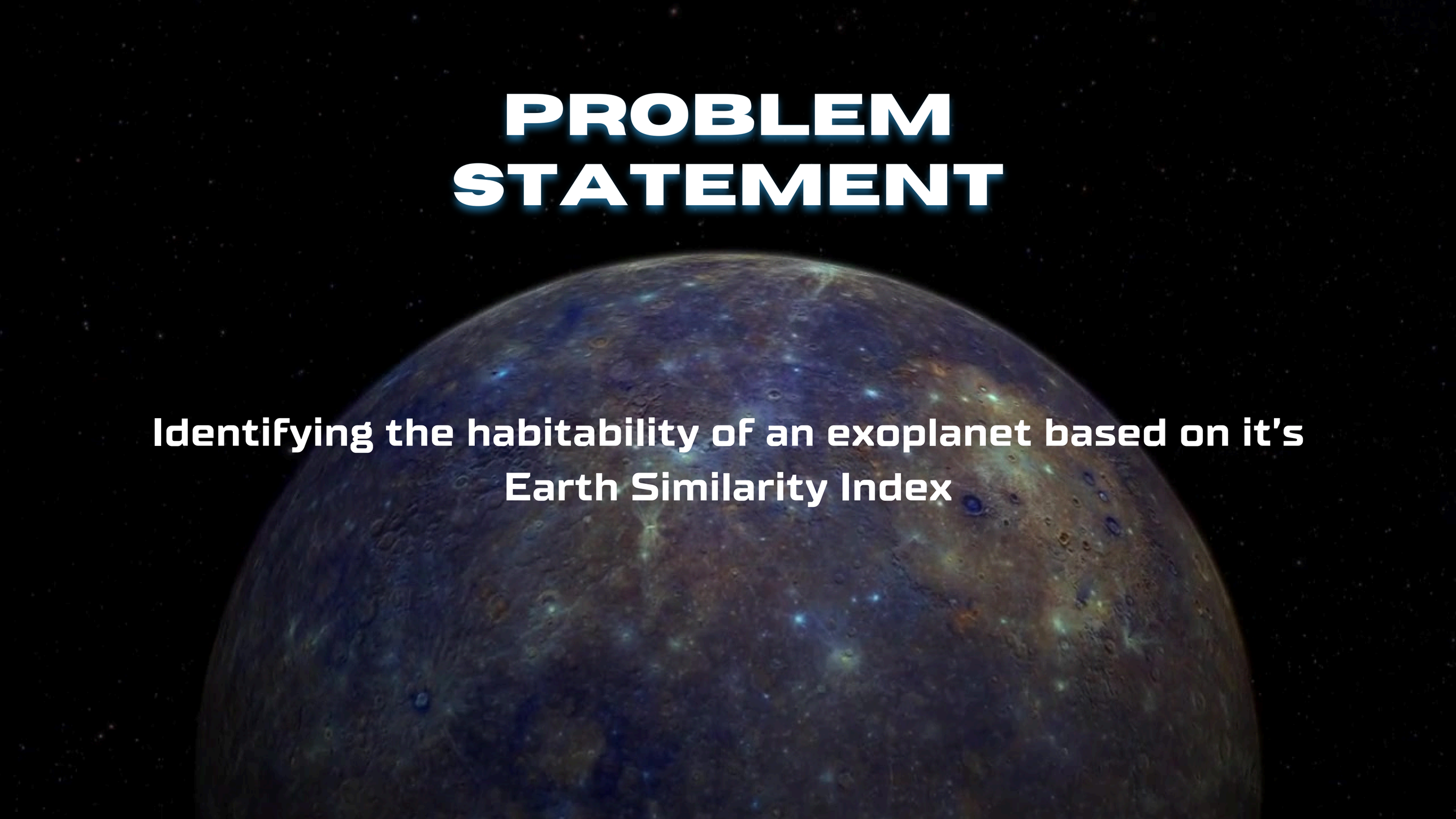
INTRODUCTION

In this project, we aim to delve into the realm of exoplanetary science, where we investigate the existence and potential habitability of planets orbiting distant stars.

We evaluate several characteristics of the exoplanets that help us determine whether an exoplanet is habitable or not based on a metric known as the Earth Similarity Index.

PROBLEM STATEMENT

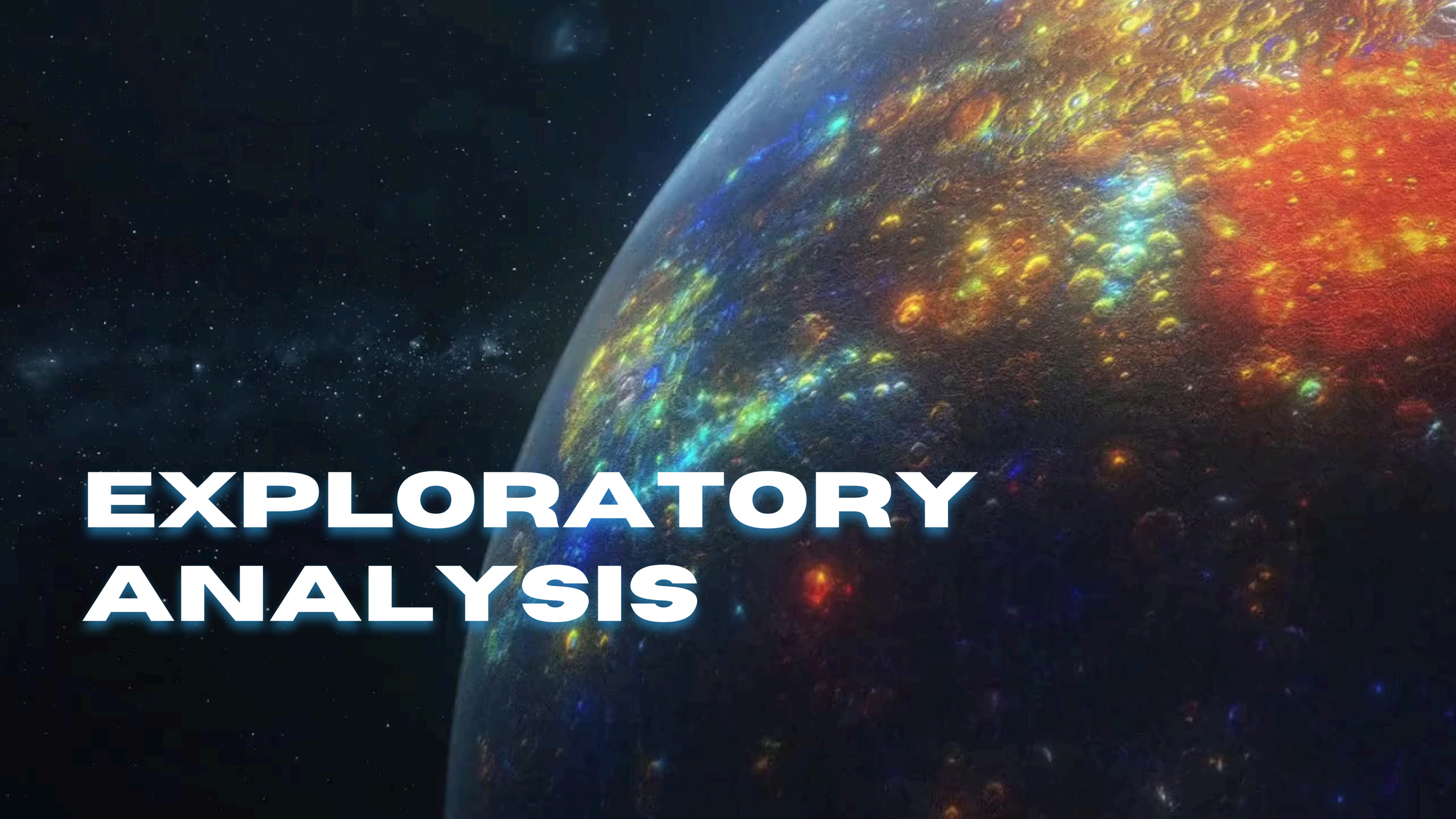


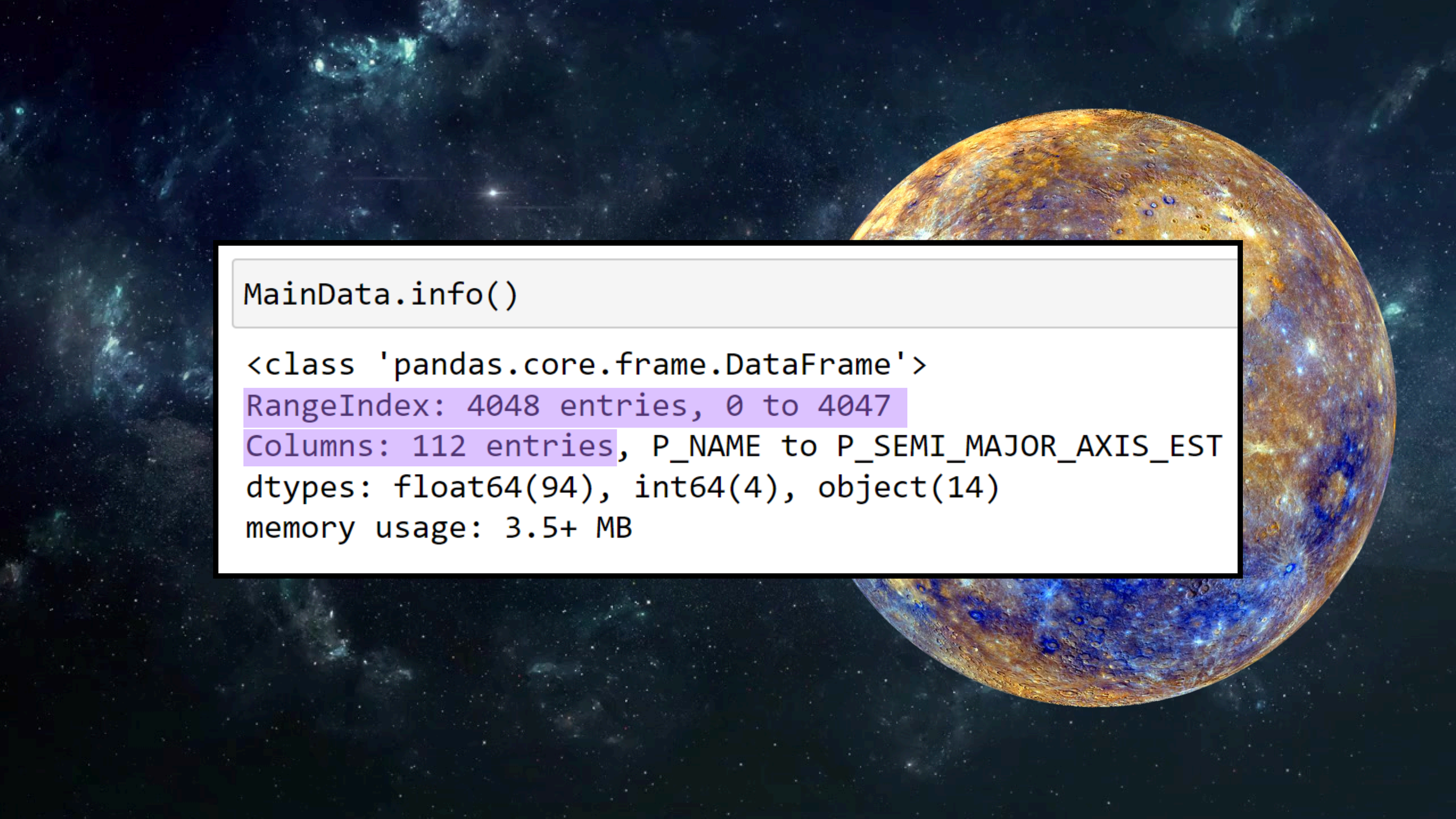
Identifying the habitability of an exoplanet based on it's Earth Similarity Index

STATISTICAL SUMMARY

- There are 4048 unique values with 112 fields.
- The response variable is ESI i.e. the Earth Similarity Index.
- There are 100 numerical variables and 12 categorical variables.

P_NAME	P_STATUS	P_MASS	P_MASS_ERROR_...	P_MASS_ERROR_...	P_RADIUS
4048 unique values	3 3	0.02 17.7k	-25k 0.27	0 26.6k	0.34 77.3
CT Cha b	3.0000000	5403.0760	-1906.9680	1906.9680	24.662000
CoRoT-1 b	3.0000000	327.36284	-38.139360	38.139360	16.702900
CoRoT-10 b	3.0000000	874.02701	-50.852480	50.852480	10.873700
CoRoT-11 b	3.0000000	740.53925	-108.06152	108.06152	16.030300
CoRoT-12 b	3.0000000	291.44828	-20.658820	22.247960	16.142400
CoRoT-13 b	3.0000000	415.71903	-20.976648	20.976648	9.9208500
CoRoT-14 b	3.0000000	2415.4928	-190.69680	190.69680	12.218900
CoRoT-16 b	3.0000000	170.03798	-26.379724	27.015380	13.115700
CoRoT-17 b	3.0000000	772.32205	-95.348401	95.348401	11.434200
CoRoT-18 b	3.0000000	1102.8632	-120.77464	120.77464	14.685100
CoRoT-19 b	3.0000000	352.78908	-19.069680	19.069680	14.460900
CoRoT-2 b	3.0000000	1102.8632	-69.922161	69.922161	16.433860
CoRoT-20 b	3.0000000	1366.6604	-63.565601	63.565601	
CoRoT-20 c	3.0000000	5403.0760	-317.82800	317.82800	
CoRoT-21 b	3.0000000	718.29129	-98.526681	98.526681	14.573000
CoRoT-22 b	3.0000000	12.077464	-8.8991841	13.984432	4.8763500
CoRoT-23 b	3.0000000	889.91841	-95.348401	95.348401	11.770500


EXPLORATORY
ANALYSIS



```
MainData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4048 entries, 0 to 4047
Columns: 112 entries, P_NAME to P_SEMI_MAJOR_AXIS_EST
dtypes: float64(94), int64(4), object(14)
memory usage: 3.5+ MB
```

```
MainData.info()
```

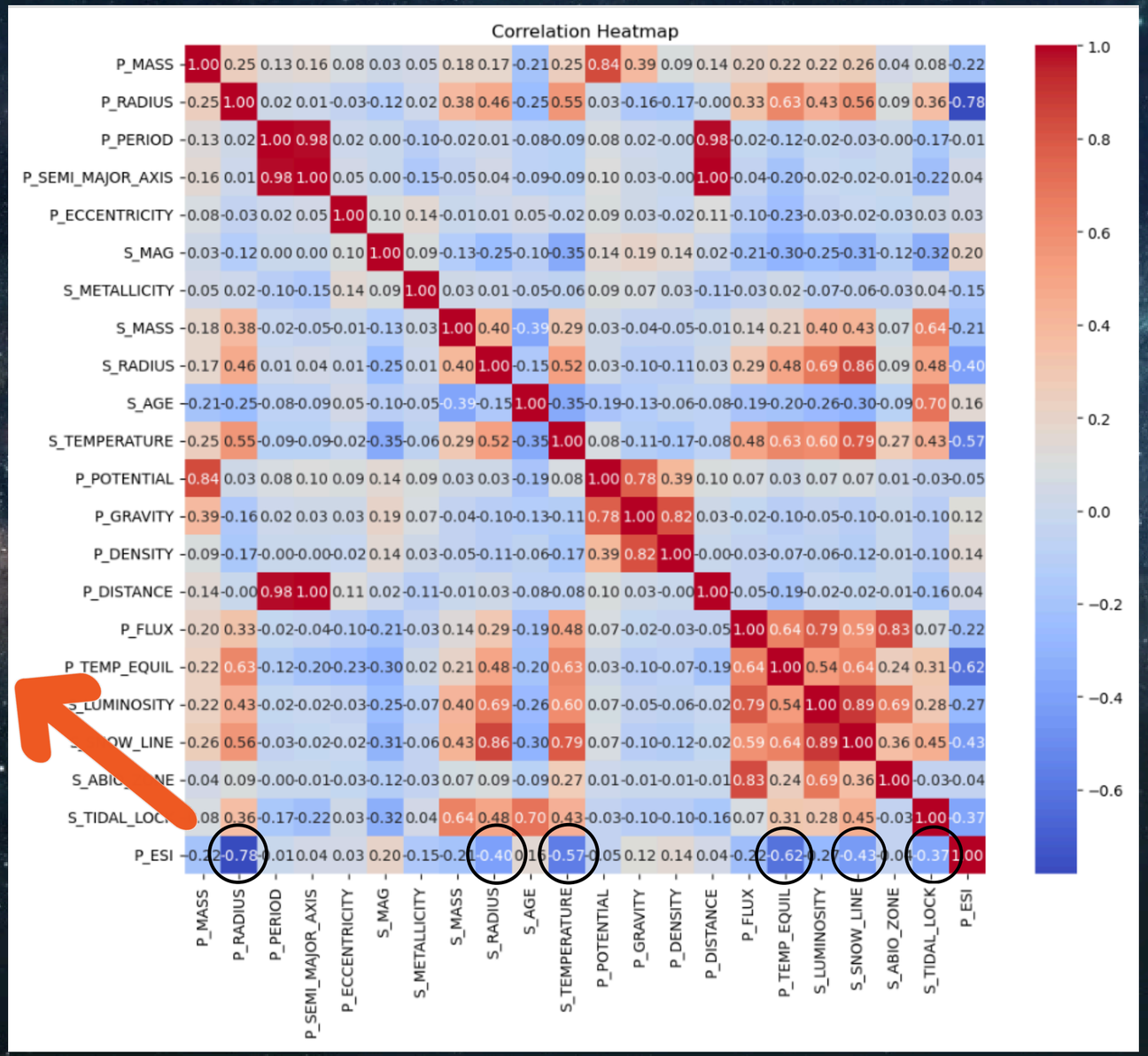
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 571 entries, 0 to 570
Data columns (total 26 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   P_NAME              571 non-null    object 
 1   P_MASS              571 non-null    float64
 2   P_RADIUS            571 non-null    float64
 3   P_PERIOD            571 non-null    float64
 4   P_SEMI_MAJOR_AXIS   435 non-null    float64
 5   P_ECCENTRICITY      474 non-null    float64
 6   S_NAME              571 non-null    object 
 7   S_MAG               511 non-null    float64
 8   S_METALLICITY       528 non-null    float64
 9   S_MASS              571 non-null    float64
 10  S_RADIUS            571 non-null    float64
 11  S_TYPE              343 non-null    object 
 12  S_AGE               341 non-null    float64
 13  S_TEMPERATURE       571 non-null    int64  
 14  P_POTENTIAL         571 non-null    float64
 15  P_GRAVITY           571 non-null    float64
 16  P_DENSITY           571 non-null    float64
 17  P_DISTANCE          571 non-null    float64
 18  P_FLUX              571 non-null    float64
 19  P_TEMP_EQUIL        571 non-null    float64
 20  P_TYPE              571 non-null    object 
 21  S_LUMINOSITY        571 non-null    float64
 22  S_SNOW_LINE         571 non-null    float64
 23  S_ABIO_ZONE         571 non-null    float64
 24  S_TIDAL_LOCK        571 non-null    float64
 25  P_ESI               571 non-null    float64
dtypes: float64(21), int64(1), object(4)
memory usage: 116.1+ KB
```

```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.8275778362284987
Mean Squared Error (MSE)        : 0.002220484374350189

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.6899154663930345
Mean Squared Error (MSE)        : 0.005234139442930869 6
```

CORRELATION MATRIX

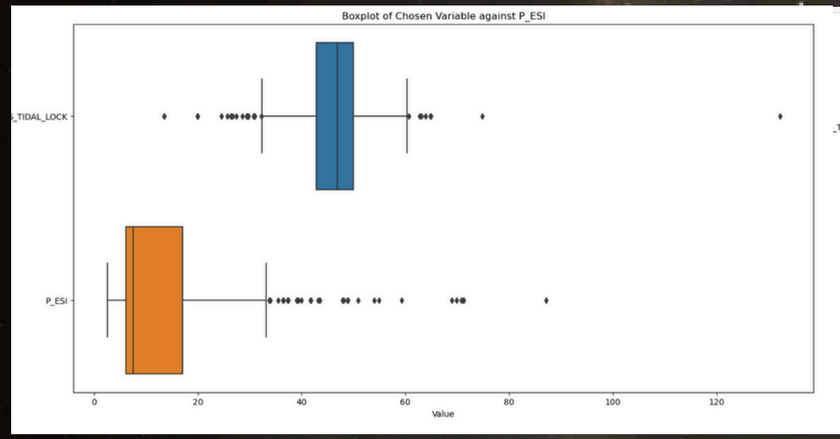
As shown on the left the 6 variables chosen are:

1.Planet radius
2.Star radius
3.Star temperature
4.Planet temperature equilibrium
5. Star Snow Line
6. Star Tidal Lock

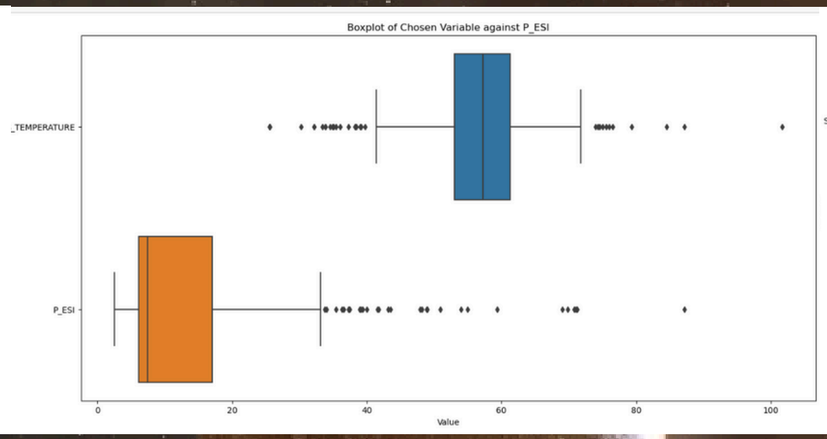


BI-VARIATE ANALYSIS

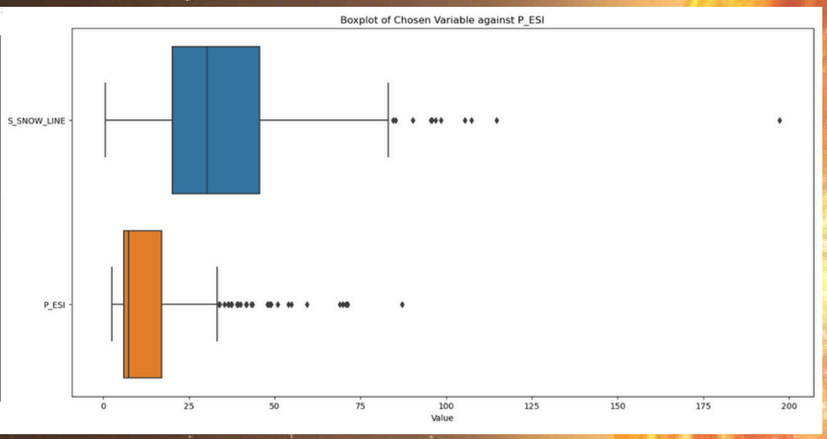
Tidal Lock



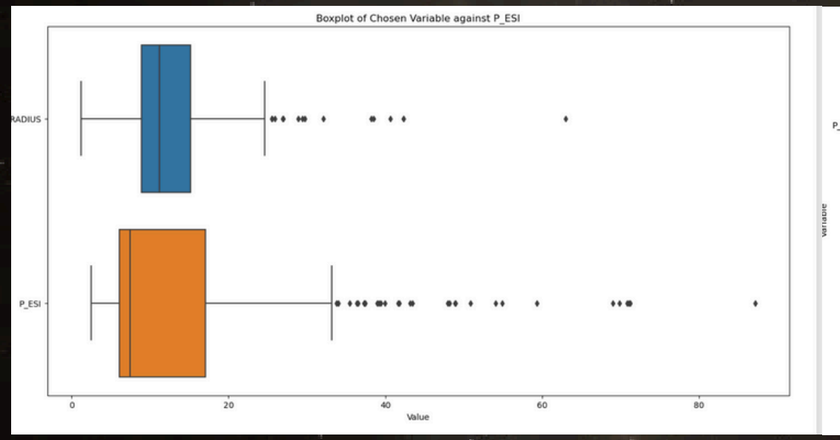
Temperature



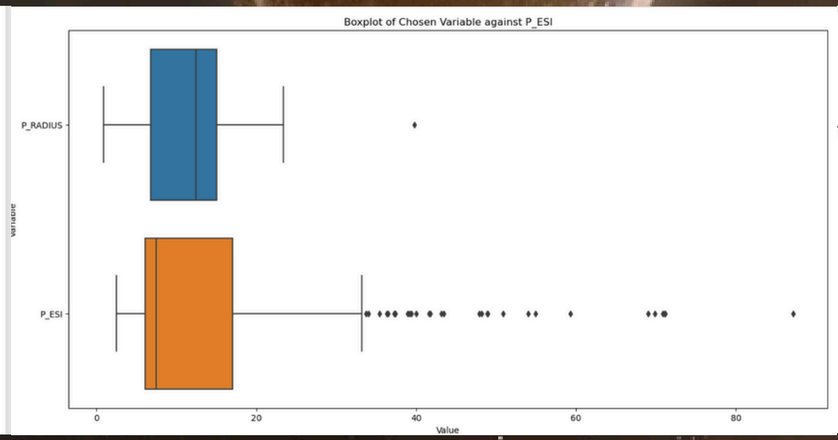
SnowLine



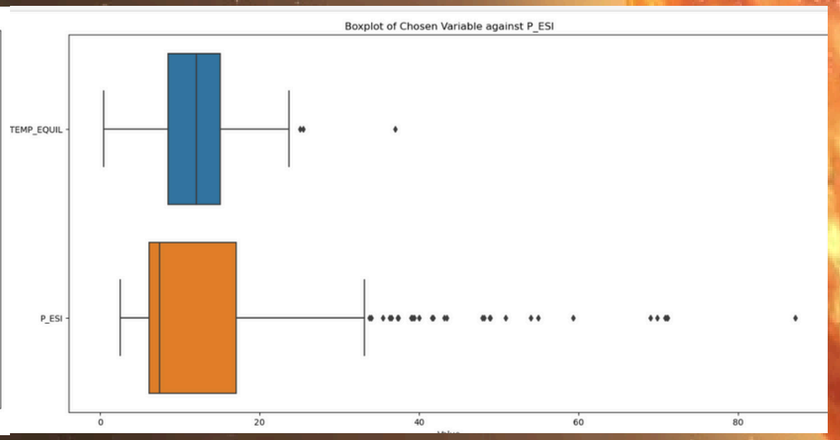
Star Radius



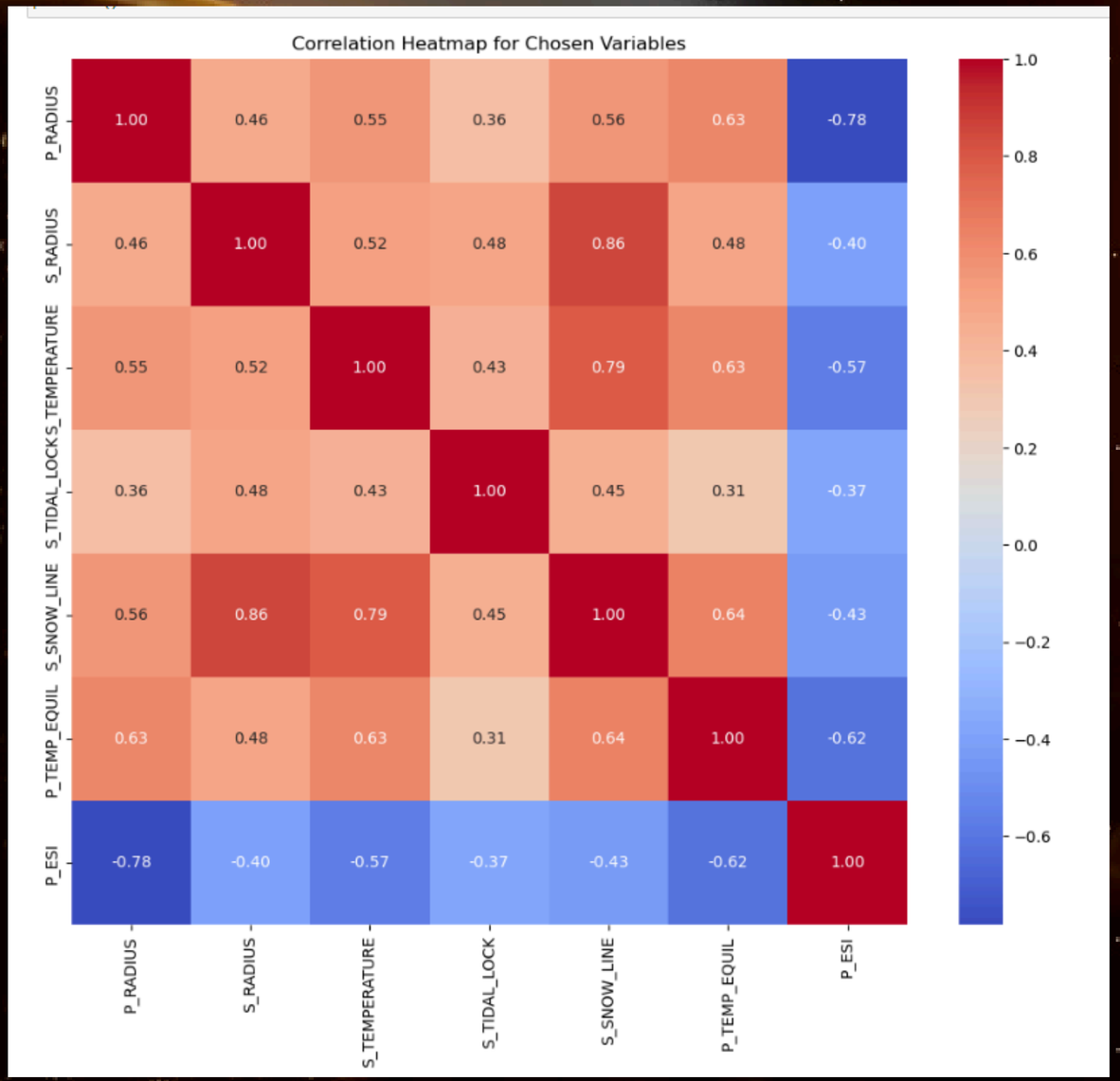
Planet Radius



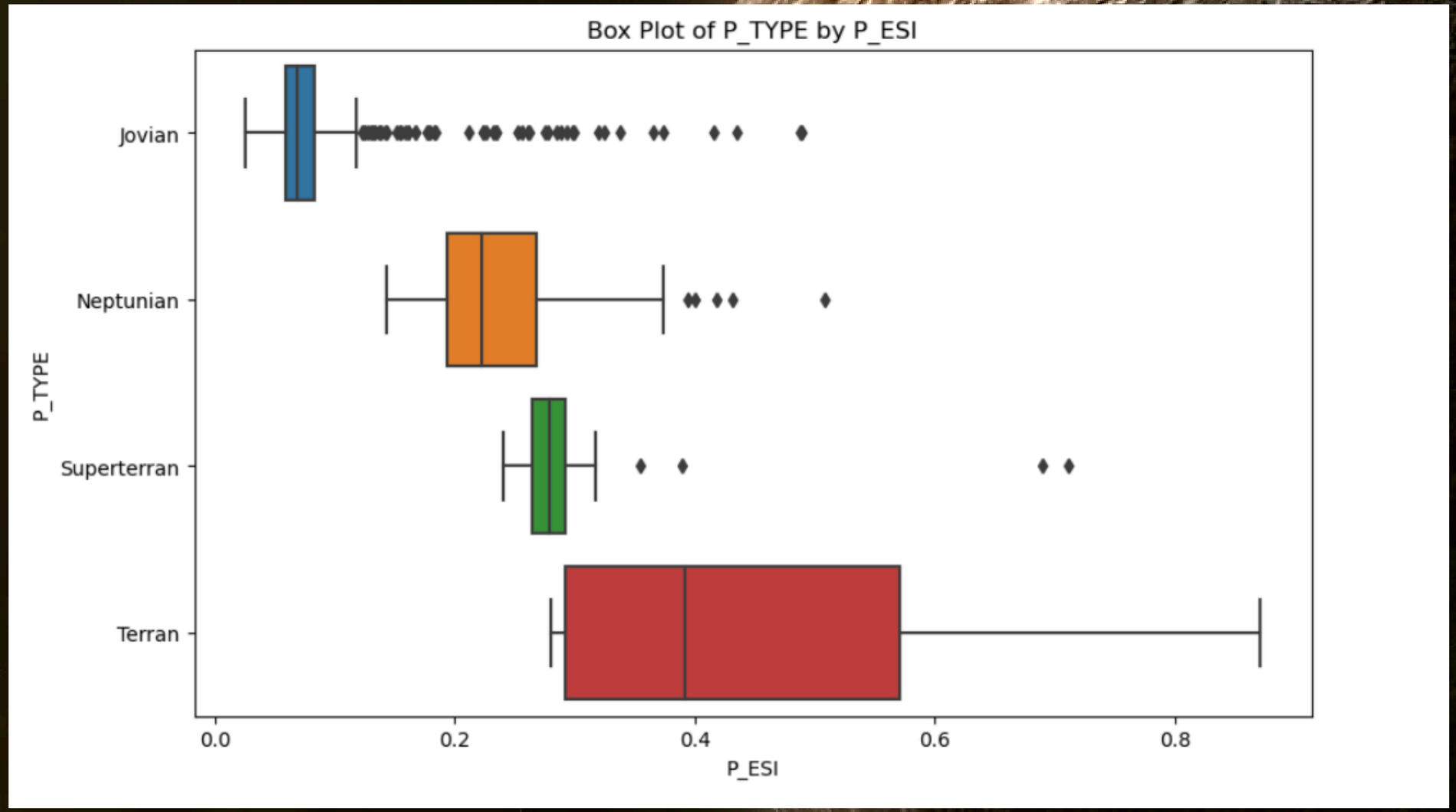
PlanetTemperatu re Equilibrium



MULTIVARIATE ANALYSIS

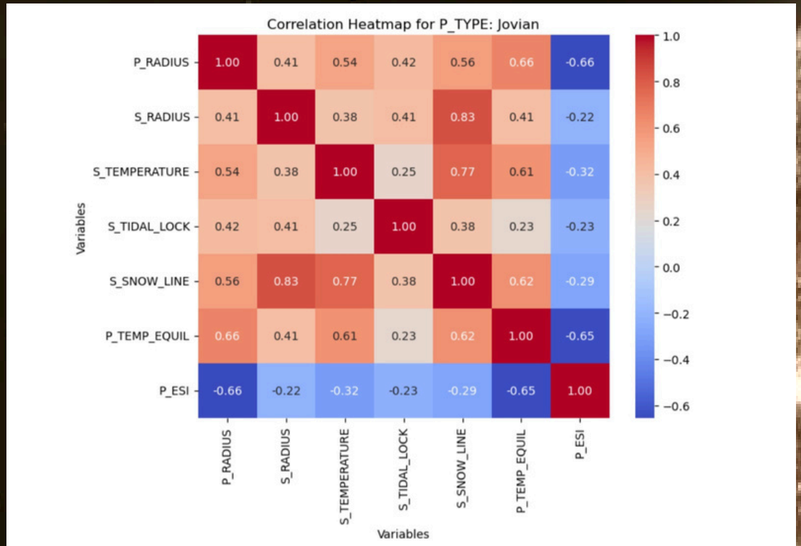


CATEGORICAL VARIABLES

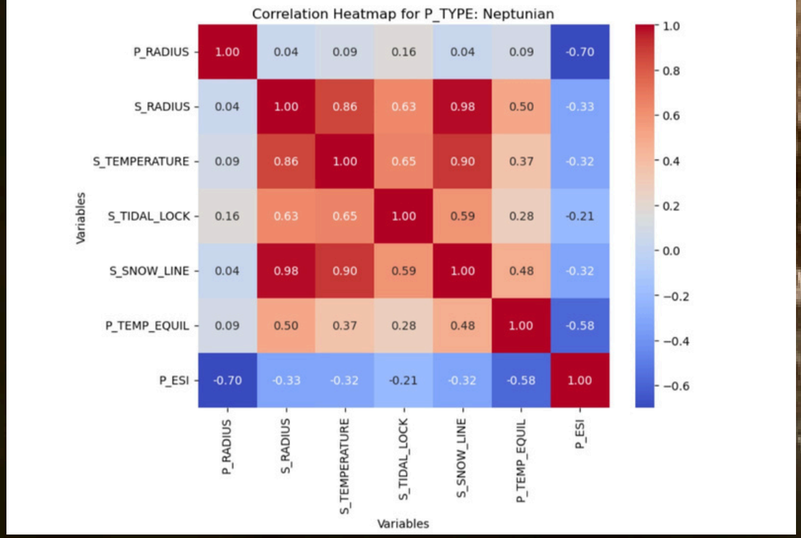


CATEGORICAL VARIABLES

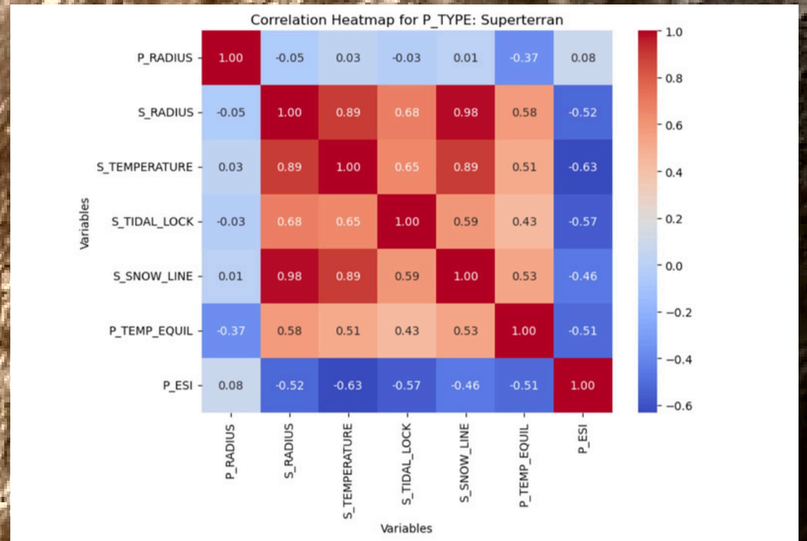
Jovian



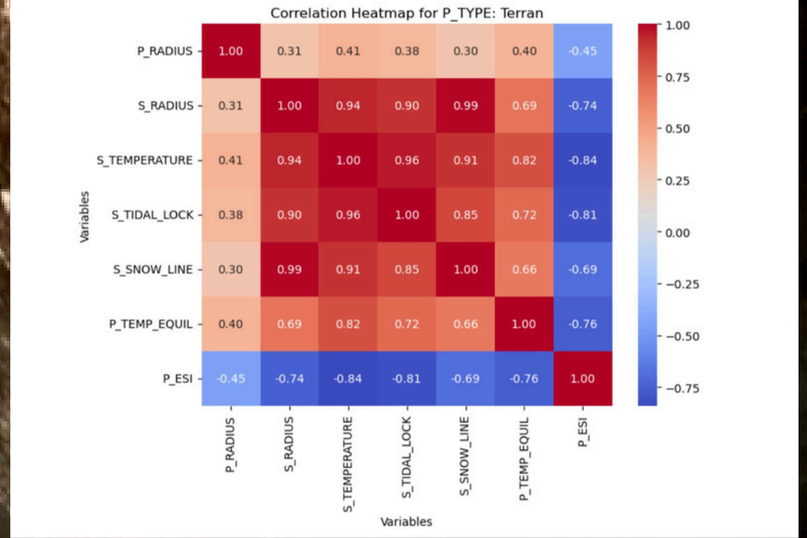
Neptunian

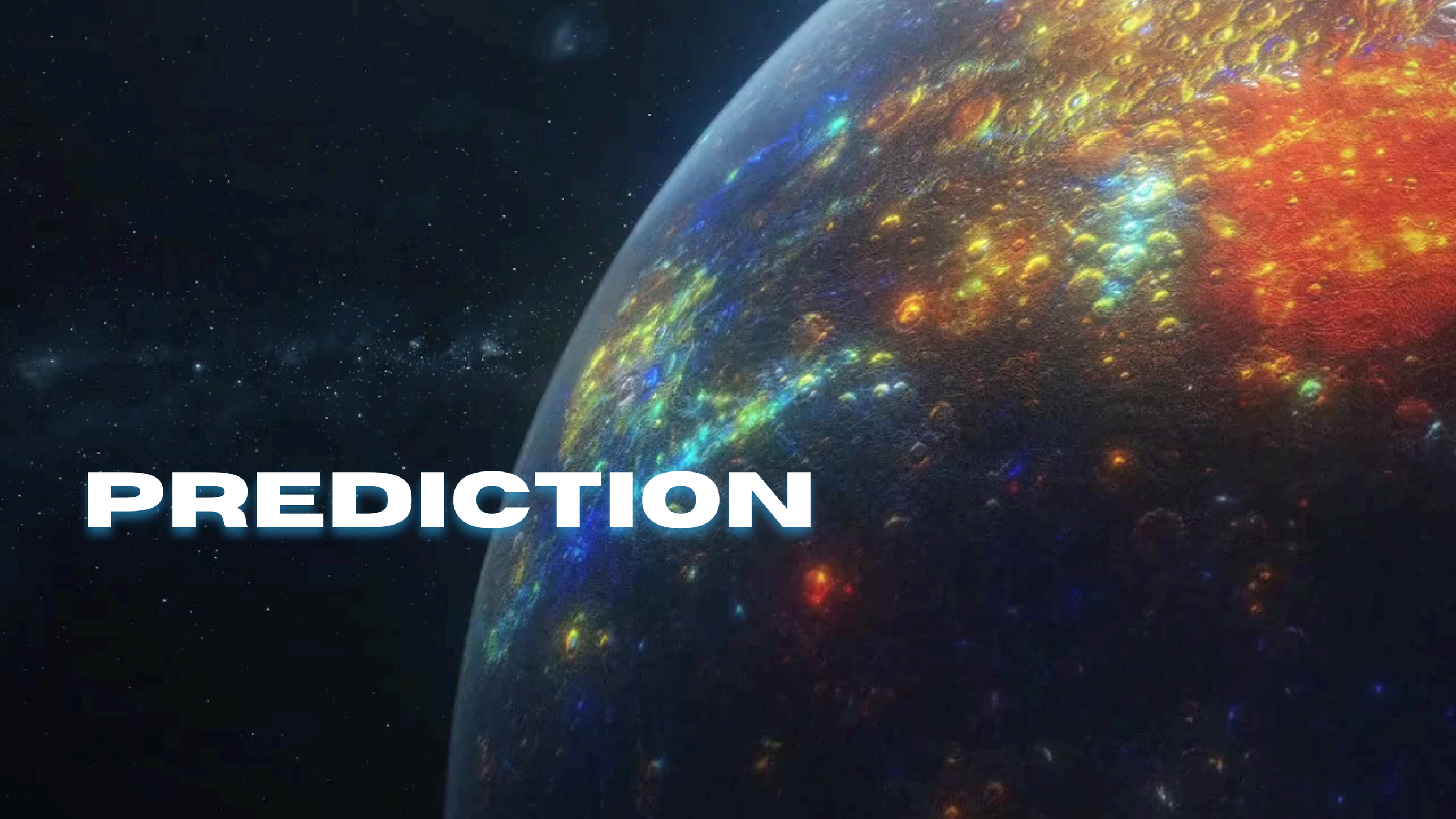


Superterran

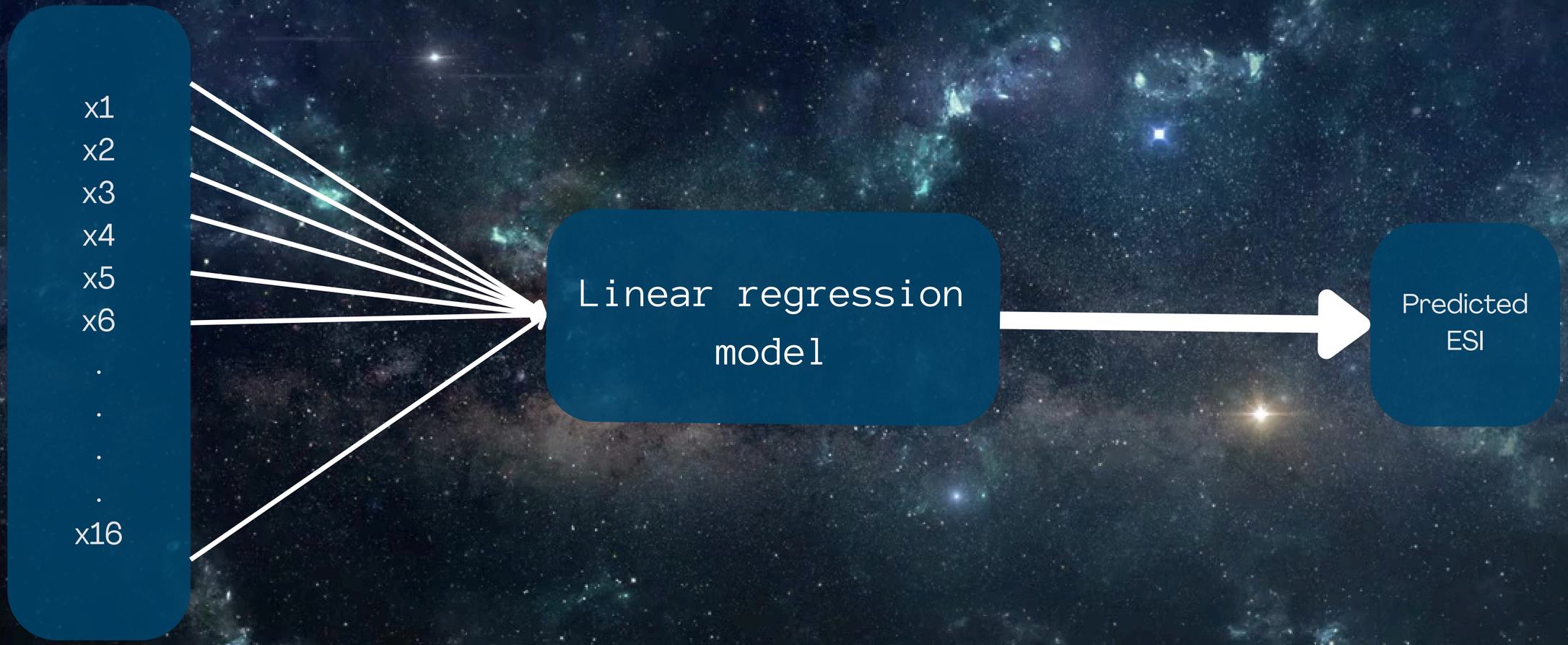


Terran



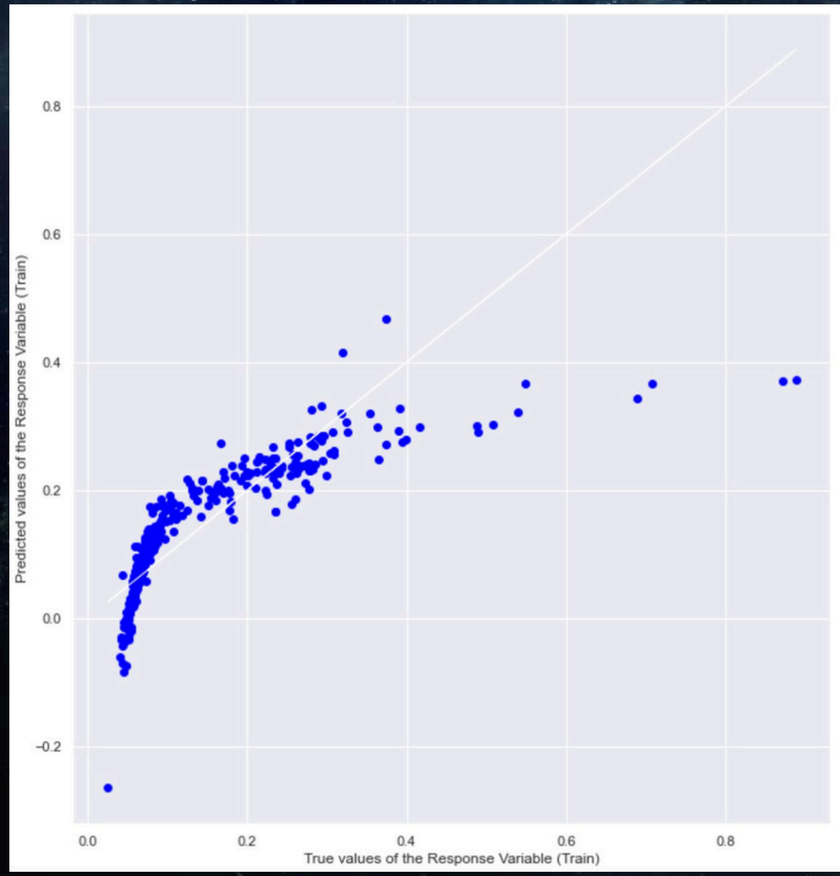

PREDICTION

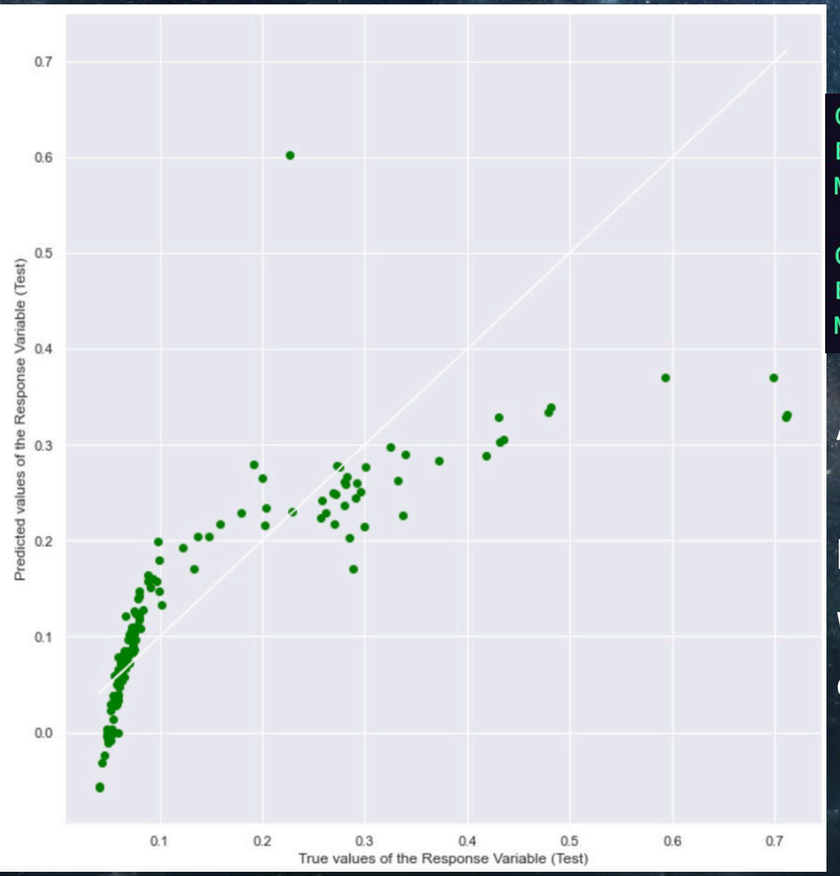
LINEAR REGRESSION



16 Numerical Variables

LINEAR REGRESSION RESULTS





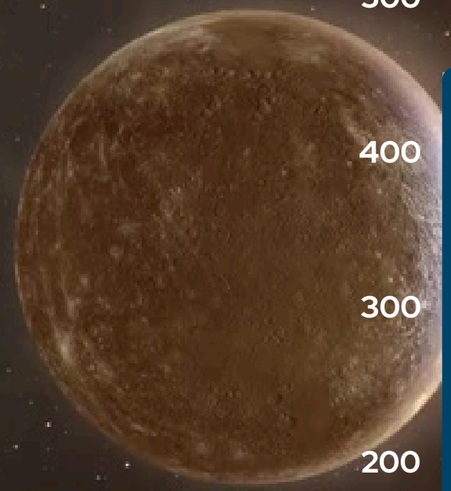

```
Goodness of Fit of Model          Train Dataset
Explained Variance (R^2)          : 0.6764723009236373
Mean Squared Error (MSE)          : 0.004288299508738812

Goodness of Fit of Model          Test Dataset
Explained Variance (R^2)          : 0.6708235901712933
Mean Squared Error (MSE)          : 0.006607670680403212
```

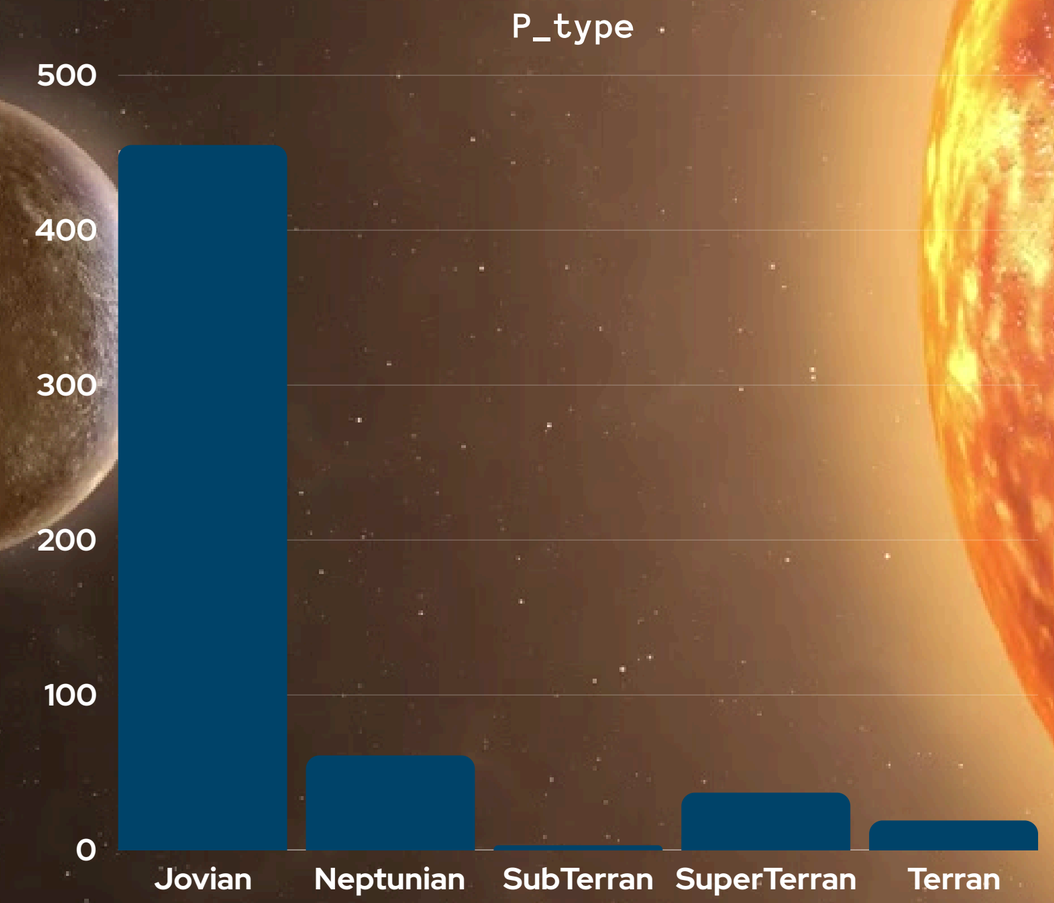
As you can see that the MSE is quite low and the Explained Variance is high so we can say that this model is quite accurate.

WHAT ABOUT PLANET TYPE?

Since Linear Regression is a numeric input to a numeric output model, we did not use the Categorical Variable “Planet Type (P_TYPE)”.

However P_TYPE is a really important variable, as we saw when we were exploring the data. So how do we implement that variable into the Linear Regression model?

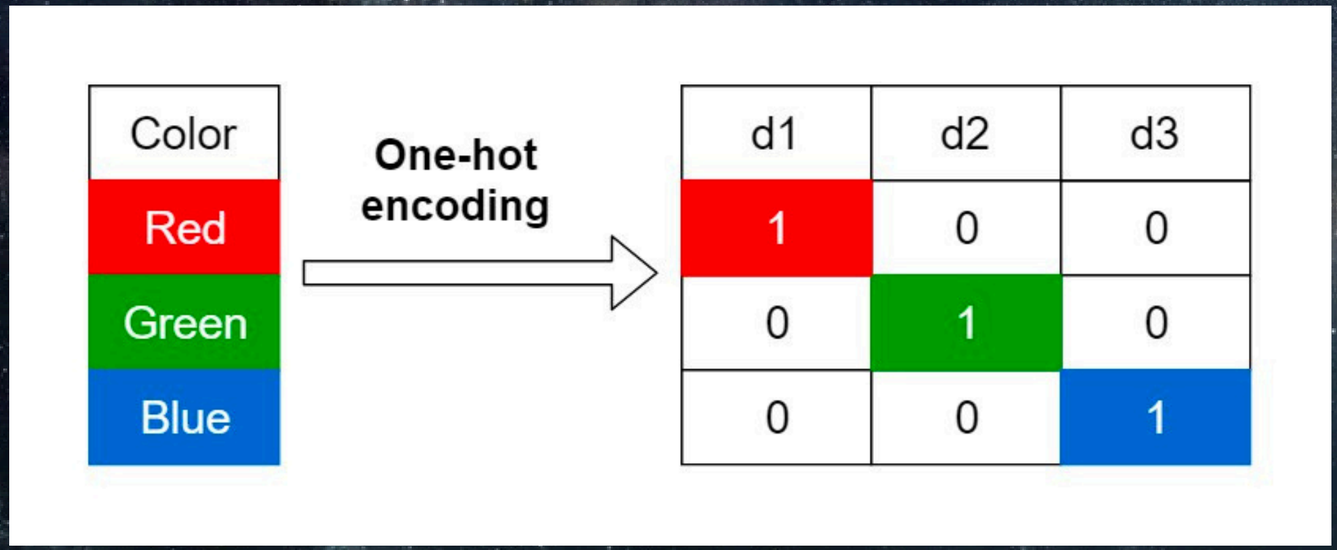




ONE HOT ENCODING

type of categories: Nominal

We use One Hot Encoding to encode the categorical information into a format that can be fed into machine learning algorithms for better results

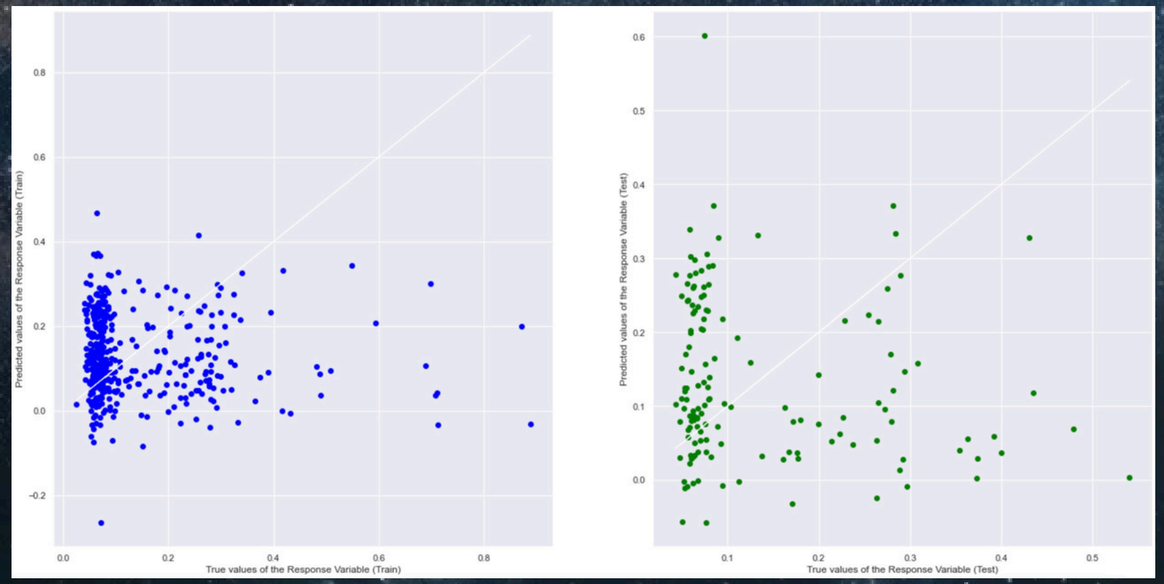


One-Hot Encoding (Integer Encoding example)

ONE HOT ENCODING REGRESSION

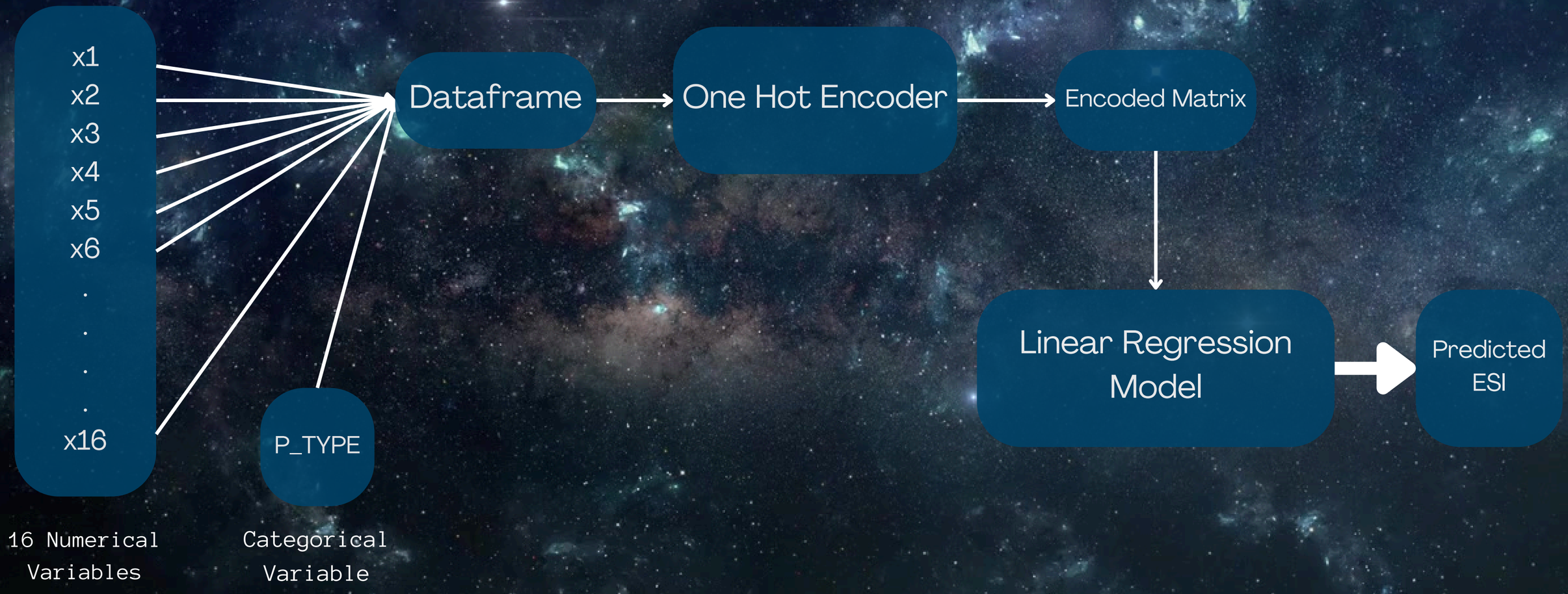
We encoded the P_TYPE variable using the one hot encoding technique and then used that variable to predict ESI variables.
Note: We are only currently using the P_TYPE variable as the input.

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import mean_squared_error
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

encoder_one_hot = OneHotEncoder()
X_train_one_hot = encoder_one_hot.fit_transform(X_train[['P_TYPE']])

model_one_hot = LinearRegression().fit(X_train_one_hot, y_train)

X_test_one_hot = encoder_one_hot.transform(X_test[['P_TYPE']])
y_pred_one_hot = model_one_hot.predict(X_test_one_hot)
mse_one_hot = mean_squared_error(y_test, y_pred_one_hot)
```

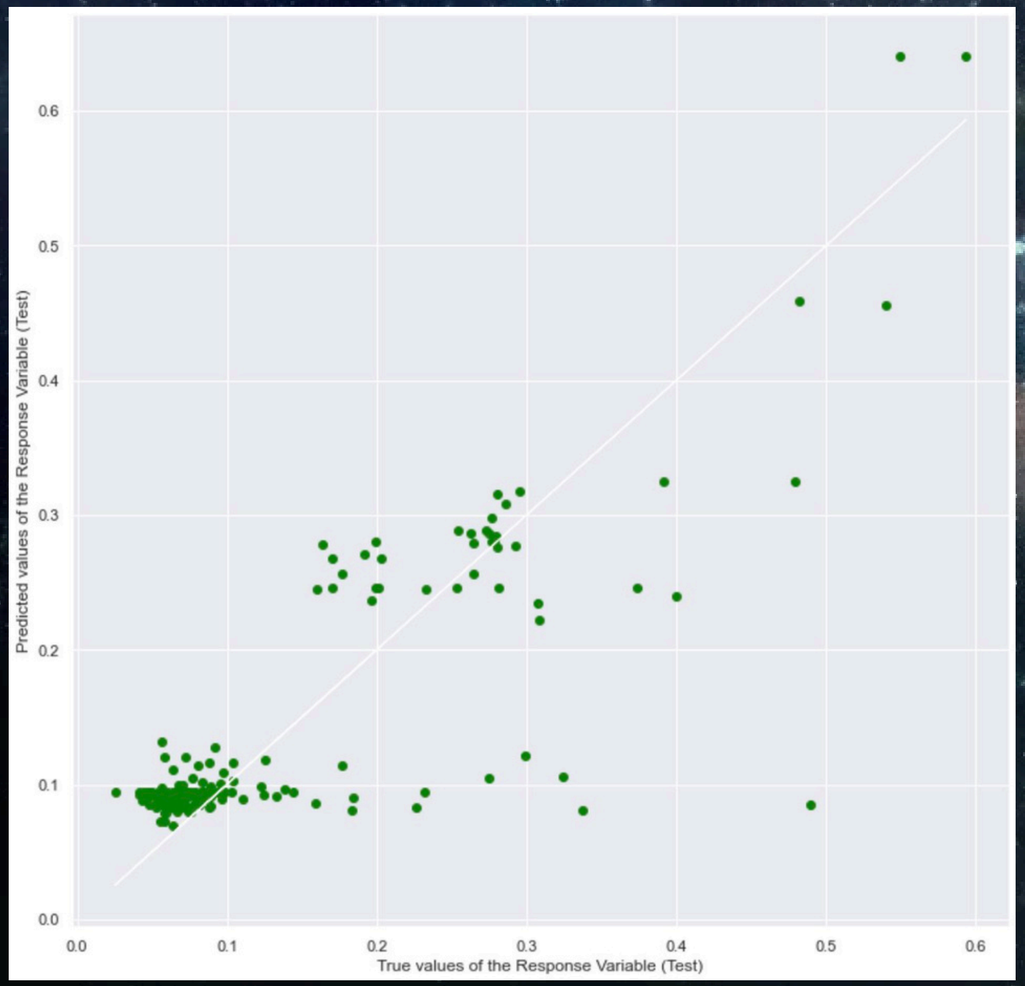


The MSE came out to be 0.0039 which suggests that this method is quite accurate for categorical to numeric prediction

ONE HOT ENCODING LINEAR REGRESSION



ONE HOT ENCODING LINEAR REGRESSION



Values received:

Mean Absolute Error =

0.038211555836443804

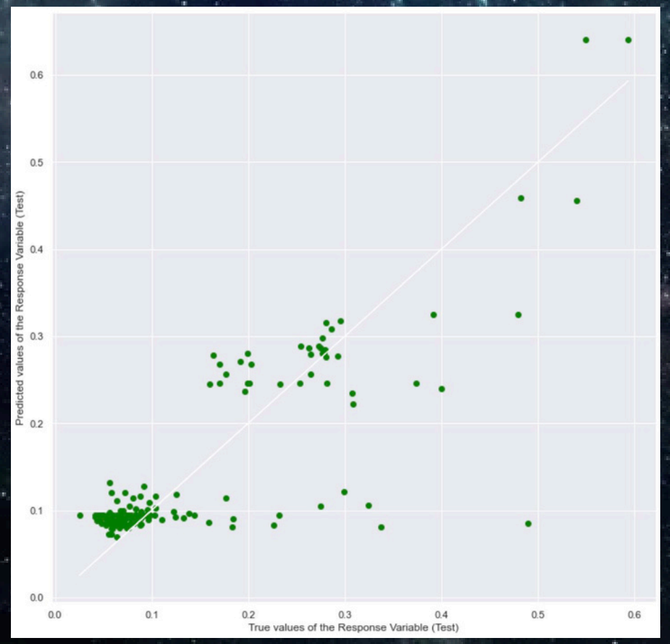
R2 score =

0.6992039976800235

This suggests that the model worked pretty well!

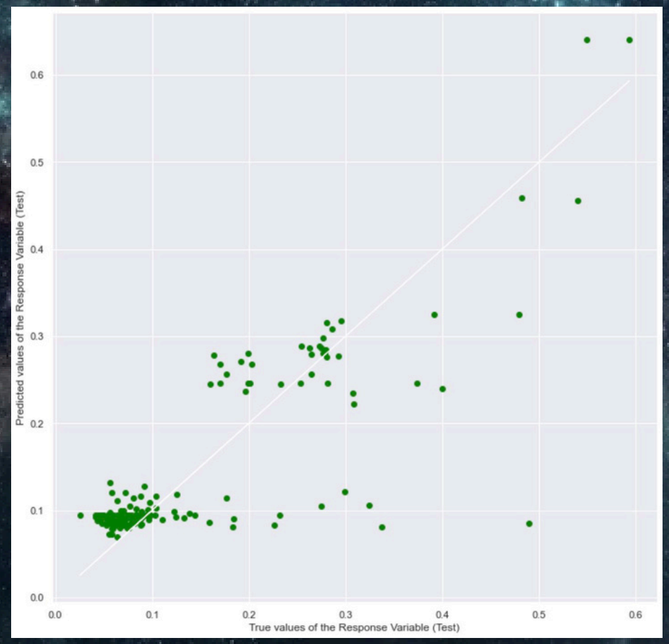
MORE WAYS TO DO THIS USING ONE HOT SEARCH

POLYNOMIAL REGRESSION



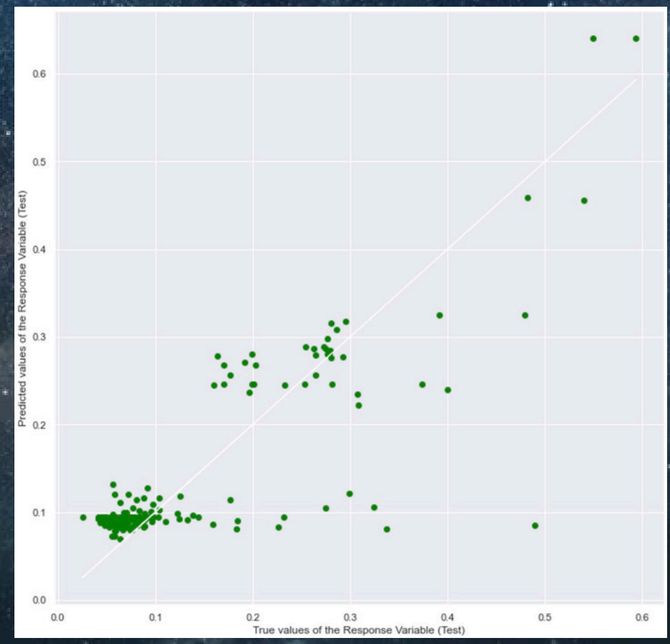
Mean Absolute Error =
0.043616479601447544
R2 score =
0.6665287540576839

DECISION TREE REGRESSION



Mean Absolute Error =
0.03287148826041665
R2 score =
0.627559294064835 2

RANDOM FOREST REGRESSION



Mean Absolute Error =
0.03287148826041665
R2 score =
0.627559294064835 2

RESULT

Determine the Earth Similarity Index of any exoplanet provided we're given the correct data.


THANK
YOU FOR
LISTENING!