INTRODUCTION

Project Title

Predicting the Habitability Potential of Exoplanets Using Machine Learning Models

Prepared By

UJJWAL KUMAR

Project Overview

- **The discovery of exoplanets has opened new possibilities in the search for life beyond Earth. However, not all discovered exoplanets are suitable for supporting life. This project focuses on predicting the habitability potential of exoplanets by analyzing scientific data using Machine Learning (ML) techniques.**
- **The main objective of this project is to preprocess raw exoplanet data and prepare it for machine learning models that can accurately classify or predict whether an exoplanet has conditions suitable for life. The project uses real-world astronomical datasets and applies data preprocessing techniques to ensure data quality, consistency, and reliability.**

● Tasks Completed

Data Preprocessing
Handling missing and inconsistent values
Feature selection and data cleaning
Data transformation for machine learning readiness

FirstNotebook

January 16, 2026

[12]:
```
#(PHASE1—FeatureSelection)
```

[1]:
```
importpandasaspd
```

[2]:
```
file=r"C:\Users\ujjwa\Downloads\PS_2025.12.30_07.05.19.csv"

withopen(file,"rb")asf:print(f.read(200))
```

```
b'# This file was produced by the NASA Exoplanet Archive
http://exoplanetarchive.ipac.caltech.edu\n# Tue Dec 30 07:05:19 2025\n#\n#
COLUMN pl_name: Planet Name\n# COLUMN hostname: Host Name\n# '
```

[3]:
```
importpandasaspd

file=r"C:\Users\ujjwa\Downloads\PS_2025.12.30_07.05.19.csv"

df=pd.read_csv(file,
    comment="#",          #IgnoreNASAheaderlines
    sep=",",engine="pyt
    hon"
)

df.head()
```

[3]:

	rowid	pl_name	hostname	pl_letter	hd_name	hip_name	tic_id	gaia_dr2_id	gaia_dr3_id	default_flag
0	1	11 Com b	11 Com	b	HD 107383	HIP 60202	TIC 72437047	Gaia DR2 3946945413106333696	Gaia DR3 3946945413106333696	1
1	2	11 Com b	11 Com	b	HD 107383	HIP 60202	TIC 72437047	Gaia DR2 3946945413106333696	Gaia DR3 3946945413106333696	0
2	3	11 Com b	11 Com	b	HD 107383	HIP 60202	TIC 72437047	Gaia DR2 3946945413106333696	Gaia DR3 3946945413106333696	0
3	4	11 UMi b	11 UMi	b	HD 136726	HIP 74793	TIC 230061010			
4	5	11 UMi b	11 UMi	b	HD 136726	HIP 74793	TIC 230061010			

3	Gaia DR2	1696798367260229376	Gaia	DR3	1696798367260229376	0

```
4   Gaia DR2 1696798367260229376  Gaia DR3 1696798367260229376                1

     ...   rowupdate  pl_pubdate  releasedate  pl_nnotes  st_nphot  st_nrvc  \
0    ...  2023-09-19     2023-08   2023-09-19        2.0       1.0      2.0
1    ...  2014-05-14     2008-01   2014-05-14        2.0       1.0      2.0
2    ...  2014-07-23     2011-08   2014-07-23        2.0       1.0      2.0
3    ...  2018-04-25     2009-10   2014-05-14        0.0       1.0      1.0
4    ...  2018-09-04     2017-03   2018-09-06        0.0       1.0      1.0

   st_nspec  pl_nespec  pl_ntranspec  pl_ndispec
0       0.0        0.0           0.0         0.0
1       0.0        0.0           0.0         0.0
2       0.0        0.0           0.0         0.0
3       0.0        0.0           0.0         0.0
4       0.0        0.0           0.0         0.0

[5 rows x 289 columns]
```

[4]:
```
df.shape
```

[4]: (39212, 289)

[5]:
```
df.columns[:10]
```

[5] :
```
Index(['rowid', 'pl_name', 'hostname', 'pl_letter', 'hd_name', 'hip_name', 'tic_id',
       'gaia_dr2_id', 'gaia_dr3_id', 'default_flag'], dtype='object')
```

[6] :
```
print(df.columns)
```

```
Index(['rowid', 'pl_name', 'hostname', 'pl_letter', 'hd_name', 'hip_name', 'tic_id',
       'gaia_dr2_id', 'gaia_dr3_id', 'default_flag',
       ...
       'rowupdate', 'pl_pubdate', 'releasedate', 'pl_nnotes', 'st_nphot', 'st_nrvc',
       'st_nspec', 'pl_nespec', 'pl_ntranspec', 'pl_ndispec'], dtype='object', length=289)
```

[7] :
```
selected_columns=["pl_r
    ade",                 #Planetradius(Earthradius)
    "pl_masse",
                          #Planetmass(Earthmass)"pl_orbper",
                          #Orbitalperiod(days)"pl_orbsmax",
                          #Semi-majoraxis(AU)"pl_eqt",
                          #Equilibriumtempe
    rature(K)"pl_dens",     #Planetdensity
    "st_teff",
                           #St
    artemperature"st_lum",
```

```
]
```

[8] :
```
core_df=df[selected_columns]
```

[9] :
```
core_df.head()
```

[9] :

	pl_rade	pl_masse	pl_orbper	pl_orbsmax	pl_eqt	pl_dens	st_teff	\
0	NaN	NaN	323.21000	1.178	NaN	NaN	4874.0	
1	NaN	NaN	326.03000	1.290	NaN	NaN	4742.0	
2	NaN	NaN	NaN	1.210	NaN	NaN	NaN	
3	NaN	NaN	516.22000	1.540	NaN	NaN	4340.0	
4	NaN	NaN	516.21997	1.530	NaN	NaN	4213.0	

	st_lum	st_met	st_spectype
0	1.97823	-0.26	G8 III
1	2.24300	-0.35	G8 III
2	NaN	NaN	NaN
3	NaN	0.04	K4 III
4	NaN	-0.02	NaN

[10] :
```
core_df.to_csv("phase1_core_features.csv",index=False)
```

[11] :
```
##(PHASE-2—DataQualityAudit)
```

[13]:
```
importpandasaspd
importmatplotlib.pyplotaspltimportseaborn
assns

df=pd.read_csv("phase1_core_features.csv")
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39212 entries, 0 to 39211 Data
columns (total 10 columns):
 #   Column       Non-Null Count   Dtype
---  -------      --------------   ------
 0   pl_rade      27015 non-null   float64
 1   pl_masse     4744 non-null    float64
 2   pl_orbper    35871 non-null   float64
 3   pl_orbsmax   21936 non-null   float64
 4   pl_eqt       17182 non-null   float64
 5   pl_dens      2713 non-null    float64
 6   st_teff      35691 non-null   float64
 7   st_lum       9642 non-null    float64
 8   st_met       24765 non-null   float64
 9   st_spectype  2890 non-null    object
dtypes: float64(9), object(1)
```

memory usage: 3.0+ MB

[15]:
```
df.describe()
```

[15]:

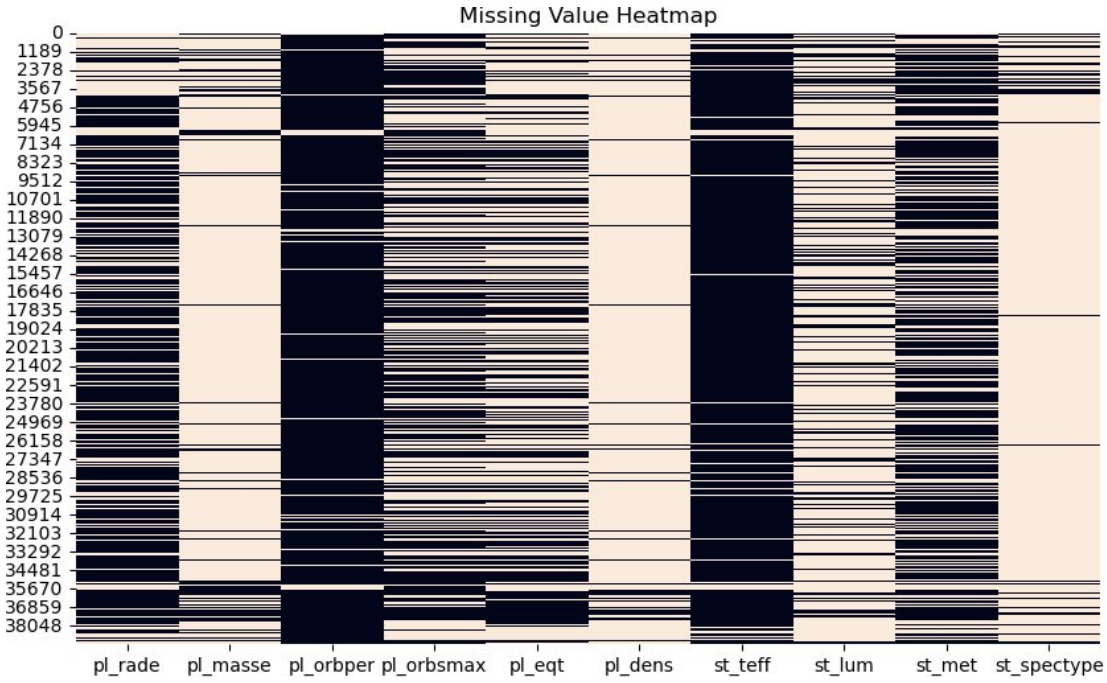
	pl_rade	pl_masse	pl_orbper	pl_orbsmax	pl_eqt	\
count	27015.000000	4744.000000	3.587100e+04	21936.000000	17182.000000	
mean	5.448188	742.251235	1.219128e+04	4.727505	880.352515	
std	71.897105	1709.000144	2.123765e+06	181.067690	428.658514	
min	0.270000	0.018000	9.070629e-02	0.004400	34.000000	
25%	1.550000	11.000000	4.396000e+00	0.054538	568.000000	
50%	2.300000	154.306465	1.033929e+01	0.100860	797.000000	
75%	3.260000	585.458751	2.665656e+01	0.219000	1104.750000	
max	4282.980000	25426.400000	4.020000e+08	19000.000000	4050.000000	

	pl_dens	st_teff	st_lum	st_met
count	2713.000000	35691.000000	9642.000000	24765.000000
mean	6.175560	5462.230067	-0.149243	-0.022364
std	65.351062	993.735003	0.718795	0.226952
min	0.000740	415.000000	-4.660000	-2.500000
25%	0.560000	5099.000000	-0.461190	-0.150000
50%	1.330000	5613.000000	-0.082860	-0.003900
75%	3.690000	5951.000000	0.304150	0.120000
max	2331.000000	57000.000000	3.260760	7.790000

[16]:
```
df.isnull().sum()
```

[16]:
```
pl_rade        12197
pl_masse       34468
pl_orbper       3341
pl_orbsmax     17276
pl_eqt         22030
pl_dens        36499
st_teff         3521
st_lum         29570
st_met         14447
st_spectype    36322
dtype: int64
```

[17]:
```
plt.figure(figsize=(10,6))sns.heatmap(df.isnu
ll(),cbar=False)
plt.title("MissingValueHeatmap")plt.show()
```



[18]:
```
df.duplicated().sum()
```

[18] : np.int64(5590)

[19] :
```
df[df.isnull().any(axis=1)].head()
```

[19] :

	pl_rade	pl_masse	pl_orbper	pl_orbsmax	pl_eqt	pl_dens	st_teff	\
0	NaN	NaN	323.21000	1.178	NaN	NaN	4874.0	
1	NaN	NaN	326.03000	1.290	NaN	NaN	4742.0	
2	NaN	NaN	NaN	1.210	NaN	NaN	NaN	
3	NaN	NaN	516.22000	1.540	NaN	NaN	4340.0	
4	NaN	NaN	516.21997	1.530	NaN	NaN	4213.0	

	st_lum	st_met	st_spectype
0	1.97823	-0.26	G8 III
1	2.24300	-0.35	G8 III
2	NaN	NaN	NaN
3	NaN	0.04	K4 III
4	NaN	-0.02	NaN

[20] :
```
##(PHASE3—MissingDataTreatment)
```

[21] :
```
importpandasaspd

df=pd.read_csv("phase1_core_features.csv")
```

[22] :
```
df.isnull().sum()
```

[22]:
```
pl_rade          12197
pl_masse         34468
pl_orbper         3341
pl_orbsmax       17276
pl_eqt           22030
pl_dens          36499
st_teff           3521
st_lum           29570
st_met           14447
st_spectype      36322
dtype: int64
```

[23]:
```
num_cols = [
    "pl_rade", "pl_masse",
    "pl_orbper",
    "pl_orbsmax",
    "pl_eqt", "pl_dens",
    "st_teff", "st_lum",
    "st_met"
]



for col in num_cols:
    df[col] = df[col].fillna(df[col].median())
```

[24]:
```
df["st_spectype"]=df["st_spectype"].fillna(df["st_spectype"].mode()[0])
```

[25]:
```
df=df.dropna(how="all")
```

[26]:
```
df.isnull().sum()
```

[26] :
```
pl_rade          0
pl_masse         0
pl_orbper        0
pl_orbsmax       0
pl_eqt           0
pl_dens          0
st_teff          0
st_lum           0
st_met           0
st_spectype      0
dtype: int64
```

[27] :
```
df.to_csv("phase3_missing_fixed.csv",index=False)
```

[28] :
```
##(PHASE4—OutlierRemoval)
```

[29] :
```
importpandasaspdimpor
tnumpyasnp

df=pd.read_csv("phase3_missing_fixed.csv")
```

[30] :
```
df=df[
    (df["pl_rade"]>0)&
    (df["pl_masse"]>0)&
    (df["pl_dens"]>0)&
    (df["pl_eqt"]>0)&(df["pl_orbsmax"]>0)
    &(df["pl_orbper"]>0)
]
```

[31] :
```
defremove_outliers_iqr(df,column):Q1=df[
    column].quantile(0.25)Q3=df[column]
    .quantile(0.75)
    IQR=Q3-Q1
    lower=Q1-1.5*IQR
    upper=Q3+1.5*IQR
    returndf[(df[column]>=lower)&(df[column]<=upper)]

forcolin["pl_rade","pl_masse","pl_dens","pl_eqt","st_teff"]:df=remove_outliers_iqr(
    df,col)
```

[32] :
```
print("Remainingplanets:",len(df))
```

Remaining planets: 16973

[33] :
```
df.to_csv("phase4_physics_clean.csv",index=False)
```

[34] :
```
##(PHASE5—UnitStandardization)
```

[35] :
```
importpandasaspd

df=pd.read_csv("phase4_physics_clean.csv")
```

[36] :
```
df=df.rename(columns={
    "pl_rade":"planet_radius_earth","pl_masse":"planet_
    mass_earth","pl_orbper":"orbital_period_days","pl_o
    rbsmax":"semi_major_axis_AU","pl_eqt":"equilibriu
    m_temp_K","pl_dens":"planet_density",
```

```
    "st_teff":"star_temp_K","st_lum":"st
    ar_luminosity","st_met":"star_metal
    licity","st_spectype":"star_type"
})
```

[37] :
```
df.head()
```

[37]:

	planet_radius_earth	planet_mass_earth	orbital_period_days	\
0	2.3	154.306465	323.210000	
1	2.3	154.306465	326.030000	
2	2.3	154.306465	10.339292	
3	2.3	154.306465	516.220000	
4	2.3	154.306465	516.219970	

	semi_major_axis_AU	equilibrium_temp_K	planet_density	star_temp_K	\
0	1.178	797.0	1.33	4874.0	
1	1.290	797.0	1.33	4742.0	
2	1.210	797.0	1.33	5613.0	
3	1.540	797.0	1.33	4340.0	
4	1.530	797.0	1.33	4213.0	

	star_luminosity	star_metallicity	star_type
0	1.97823	-0.2600	G8 III
1	2.24300	-0.3500	G8 III
2	-0.08286	-0.0039	G0 V
3	-0.08286	0.0400	K4 III
4	-0.08286	-0.0200	G0 V

[38] :
```
df.to_csv("phase5_standardized.csv",index=False)
```

[39] :
```
##(PHASE6—FeatureEngineering)
```

[40] :
```
importpandasaspdimpor
tnumpyasnp

df=pd.read_csv("phase5_standardized.csv")
```

[41] :
```
earth_temp=288
earth_radius=1
earth_distance=1
earth_lum=1

df["habitability_score"]=(
    (1-abs(df["equilibrium_temp_K"]-earth_temp)/earth_temp)+(1-
    abs(df["planet_radius_earth"]-earth_radius))+
    (1-abs(df["semi_major_axis_AU"]-earth_distance))+
```

```
    (1-abs(df["star_luminosity"]-earth_lum))
)/4
```

[42] :
```
sun_temp=5778

df["stellar_compatibility"]=1-abs(df["star_temp_K"]-sun_temp)/sun_temp
```

[43] :
```
df["orbital_stability"]=1-abs(df["semi_major_axis_AU"]-1)
```

[44] :
```
df["habitability_score"]=df["habitability_score"].clip(0,1)
df["stellar_compatibility"]=df["stellar_compatibility"].clip(0,1)
df["orbital_stability"]=df["orbital_stability"].clip(0,1)
```

[45] :
```
df[["habitability_score","stellar_compatibility","orbital_stability"]].head()
```

[45] :

	habitability_score	stellar_compatibility	orbital_stability
0	0.0	0.843544	0.822
1	0.0	0.820699	0.710
2	0.0	0.971443	0.790
3	0.0	0.751125	0.460
4	0.0	0.729145	0.470

[46] :
```
df.to_csv("phase6_engineered.csv",index=False)
```

[47] :
```
##(PHASE7—CategoricalEncoding)
```

[48] :
```
importpandasaspd

df=pd.read_csv("phase6_engineered.csv")
```

[49] :
```
df["star_type"].value_counts().head()
```

[49] :
```
star_type
G0 V        15853
G5             54
K0 III         49
G5 V           49
K0 V           37
Name: count, dtype: int64
```

[50] :
```
df["star_class"]=df["star_type"].str[0]
```

[51] :
```
star_encoded=pd.get_dummies(df["star_class"],prefix="star")df=pd.concat([df,
star_encoded],axis=1)
```

[52] :
```
df=df.drop(columns=["star_type","star_class"])
```

[53] :
```
df.to_csv("phase7_encoded.csv",index=False)
```

[54] :
```
##(PHASE8—FeatureScaling)
```

[55] :
```
importpandasaspd
fromsklearn.preprocessingimportMinMaxScaler

df=pd.read_csv("phase7_encoded.csv")
```

[56] :
```
scale_columns=["planet_radius_earth","plan
    et_mass_earth","orbital_period_days","
    semi_major_axis_AU","equilibrium_tem
    p_K","planet_density","star_temp_K","st
    ar_luminosity","star_metallicity","habit
    ability_score","stellar_compatibility","o
    rbital_stability"
]
```

[57] :
```
scaler=MinMaxScaler()
df[scale_columns]=scaler.fit_transform(df[scale_columns])
```

[58] :
```
df[scale_columns].describe()
```

[58]:

	planet_radius_earth	planet_mass_earth	orbital_period_days	\
count	16973.000000	16973.0	16973.000000	
mean	0.476331	0.0	0.000908	
std	0.149807	0.0	0.010222	
min	0.000000	0.0	0.000000	
25%	0.502742	0.0	0.000032	
50%	0.502742	0.0	0.000060	
75%	0.502742	0.0	0.000159	
max	1.000000	0.0	1.000000	

	semi_major_axis_AU	equilibrium_temp_K	planet_density	star_temp_K	\
count	16973.000000	16973.0	16973.0	16973.000000	
mean	0.003963	0.0	0.0	0.523069	
std	0.016258	0.0	0.0	0.176383	
min	0.000000	0.0	0.0	0.000000	
25%	0.001364	0.0	0.0	0.419277	
50%	0.001364	0.0	0.0	0.554561	
75%	0.001364	0.0	0.0	0.640964	
max	1.000000	0.0	0.0	1.000000	

	star_luminosity	star_metallicity	habitability_score	\
count	16973.000000	16973.000000	16973.000000	
mean	0.357683	0.632034	0.009735	
std	0.065316	0.087595	0.052288	
min	0.000000	0.000000	0.000000	
25%	0.350352	0.622563	0.000000	
50%	0.350352	0.622563	0.000000	
75%	0.350352	0.656250	0.000000	
max	1.000000	1.000000	1.000000	

	stellar_compatibility	orbital_stability
count	16973.000000	16973.000000
mean	0.760628	0.124483
std	0.216317	0.125985
min	0.000000	0.000000
25%	0.655405	0.100860
50%	0.839527	0.100860
75%	0.907095	0.100860
max	1.000000	1.000000

[59] :
```
df.to_csv("phase8_scaled.csv",index=False)
```

[60] :
```
##(PHASE9—TargetVariable)
```

[61] :
```
importpandasaspd

df=pd.read_csv("phase8_scaled.csv")
```

[62] :
```
df["target_habitable"]=( (df["habitability_score"]>0.6)&(
    df["stellar_compatibility"]>0.5)&(df["orbital_stabili
    ty"]>0.5)
).astype(int)
```

[63] :
```
df["target_habitable"].value_counts()
```

[63] :
```
target_habitable
0    16952
1       21
Name: count, dtype: int64
```

[64] :
```
df.to_csv("phase9_labeled.csv",index=False)
```

[65] :
```
##(PHASE10—FinalOutput)
```

[66] :
```
importpandasaspd
```

```
df=pd.read_csv("phase9_labeled.csv")
```

[67] :
```
drop_cols=[
    "habitability_score","stellar_compatibilit
    y","orbital_stability"
]

df=df.drop(columns=drop_cols)
```

[68] :
```
df.to_csv("preprocessed.csv",index=False)
```

[ ]:
```

```