IN PROGRESS - Northwestern MSDS-AI Capstone - Shane J. Robinson

Clustering NASA's Exoplanet Archive With K-Means to Find New-Earth Candidates

In [1]:
```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_samples, silhouette_score
import plotly.express as px
import plotly.graph_objects as go
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
### load exoplanet dataset ###
df = pd.read_excel('/Users/shaner/Desktop/nasa_exoplanets.xlsx')
df.head()
```

Out[2]:

	pl_name	hostname	pl_orbper	pl_orbpererr1	pl_orbpererr2	pl_orbperlim	pl_orbsmax	pl_orbsmaxerr1	pl_orbsmaxerr2	pl_orbsmaxlim	...	pl_masseerr2	pl_masselim	pl_orbeccen	pl_orbeccenerr1
0	11 Com b	11 Com	NaN	NaN	NaN	NaN	1.21	0.06	-0.05	0.0	...	NaN	NaN	NaN	NaN
1	11 Com b	11 Com	326.03000	0.32	-0.32	0.0	1.29	0.05	-0.05	0.0	...	NaN	NaN	0.231	0.005
2	11 UMi b	11 UMi	NaN	NaN	NaN	NaN	1.51	0.06	-0.05	0.0	...	NaN	NaN	NaN	NaN
3	11 UMi b	11 UMi	516.21997	3.20	-3.20	0.0	1.53	0.07	-0.07	0.0	...	NaN	NaN	0.080	0.030
4	11 UMi b	11 UMi	516.22000	3.25	-3.25	0.0	1.54	0.07	-0.07	0.0	...	NaN	NaN	0.080	0.030

5 rows × 26 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34979 entries, 0 to 34978
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   pl_name          34979 non-null  object 
 1   hostname         34979 non-null  object 
 2   pl_orbper        31918 non-null  float64
 3   pl_orbpererr1    30584 non-null  float64
 4   pl_orbpererr2    30583 non-null  float64
 5   pl_orbperlim     31918 non-null  float64
 6   pl_orbsmax       19324 non-null  float64
 7   pl_orbsmaxerr1   5031 non-null   float64
 8   pl_orbsmaxerr2   5030 non-null   float64
 9   pl_orbsmaxlim    22103 non-null  float64
 10  pl_rade          24108 non-null  float64
 11  pl_radeerr1      23387 non-null  float64
 12  pl_radeerr2      23387 non-null  float64
 13  pl_radelim       26853 non-null  float64
 14  pl_masse         3515 non-null   float64
 15  pl_masseerr1     3264 non-null   float64
 16  pl_masseerr2     3264 non-null   float64
 17  pl_masselim      3547 non-null   float64
 18  pl_orbeccen      17533 non-null  float64
 19  pl_orbeccenerr1  3011 non-null   float64
 20  pl_orbeccenerr2  3010 non-null   float64
 21  pl_orbeccenlim   20279 non-null  float64
 22  pl_eqt           16029 non-null  float64
 23  pl_eqterr1       1793 non-null   float64
 24  pl_eqterr2       1793 non-null   float64
 25  pl_eqtlim        18774 non-null  float64
dtypes: float64(24), object(2)
memory usage: 6.9+ MB
```

In [4]:
```
### drop rows with null values ###
df.dropna(inplace=True)

### keep only relevant columns ###
columns_to_keep = ['pl_name','hostname','pl_orbper','pl_orbsmax',
                   'pl_rade','pl_masse','pl_orbeccen','pl_eqt']
df = df[columns_to_keep]

df.head()
```

Out[4]:

	pl_name	hostname	pl_orbper	pl_orbsmax	pl_rade	pl_masse	pl_orbeccen	pl_eqt
103	55 Cnc e	55 Cnc	0.736544	0.01544	2.080	7.8100	0.061	1958.0
205	CoRoT-1 b	CoRoT-1	1.508977	0.02590	15.917	340.0781	0.071	1834.0
213	CoRoT-10 b	CoRoT-10	13.240600	0.10550	10.870	874.0000	0.530	600.0
222	CoRoT-12 b	CoRoT-12	2.828042	0.04016	16.140	291.4380	0.070	1442.0
248	CoRoT-19 b	CoRoT-19	3.897130	0.05180	14.460	352.7800	0.047	2000.0

In [5]:
```
### new row of data for earth ###
new_row = {'pl_name': 'Earth', 'hostname': 'Sol','pl_orbper': 365,'pl_orbsmax': 1,
           'pl_rade': 1,'pl_masse': 1,'pl_orbeccen': 0.0167, 'pl_eqt': 255}

### add the new row using the loc indexer ###
df.loc[len(df)] = new_row

df.tail()
```

Out[5]:

	pl_name	hostname	pl_orbper	pl_orbsmax	pl_rade	pl_masse	pl_orbeccen	pl_eqt
34703	WASP-89 b	WASP-89	3.356423	0.04270	11.657	1875.19700	0.1930	1120.0
34783	Wolf 503 b	Wolf 503	6.001270	0.05712	2.043	6.27000	0.4090	789.0
34785	Wolf 503 b	Wolf 503	6.001270	0.05706	2.043	6.26000	0.4100	790.0
34854	XO-7 b	XO-7	2.864142	0.04421	15.390	225.34147	0.0380	1743.0
328	Earth	Sol	365.000000	1.00000	1.000	1.00000	0.0167	255.0

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 329 entries, 103 to 328
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   pl_name      329 non-null    object 
 1   hostname     329 non-null    object 
 2   pl_orbper    329 non-null    float64
 3   pl_orbsmax   329 non-null    float64
 4   pl_rade      329 non-null    float64
 5   pl_masse     329 non-null    float64
 6   pl_orbeccen  329 non-null    float64
 7   pl_eqt       329 non-null    float64
dtypes: float64(6), object(2)
memory usage: 23.1+ KB
```

In [7]:
```
### select features for clustering and pca ###
features = ['pl_orbper','pl_orbsmax','pl_rade','pl_masse','pl_orbeccen','pl_eqt']
```

In [8]:
```
### standardize the data ###
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[features])
```

In [9]:
```
### perform pca ###
pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_data)
```

In [10]:
```
### calculate explained variance ratio and cumulative explained variance ###
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_explained_variance = np.cumsum(explained_variance_ratio)
```

In [11]:
```
print(explained_variance_ratio)
```

```
[0.44611392 0.25580946]
```

In [12]:
```
print(cumulative_explained_variance)
```

```
[0.44611392 0.70192338]
```

In [13]:
```
### add pca results to the original dataframe ###
df['PC1'] = pca_result[:, 0]
df['PC2'] = pca_result[:, 1]
```

In [ ]:
```

```