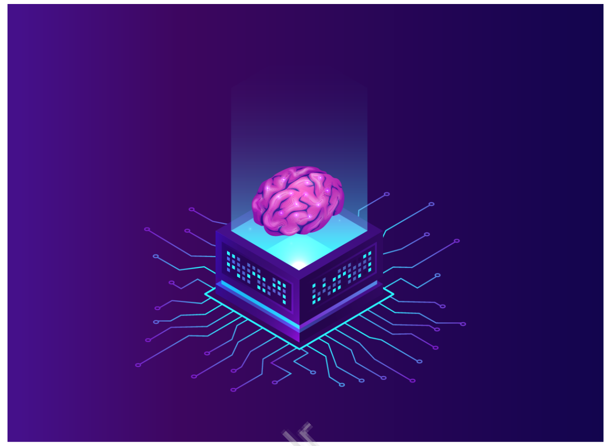
PRO-C132

Data-Science 2



What is our GOAL for this MODULE?

The goal of this module is to filter more exo-planets by applying statistics and machine learning.

What did we ACHIEVE in the class TODAY?

- We plotted different charts in this class
- We learned about different types of planets
- We applied K-Means Clustering Algorithm on the Data to find insights

Which CONCEPTS/CODING BLOCKS did we cover today?

- Plotly
- Python
- Logical reasoning
- K-Means
- Matplotlib

How did we DO the activities?

1. Let's start from where we left it last time. We had a variable **low_gravity_planets** containing the list of all the planets with gravity less than **100m/s.**
2. We observe the headers and see that there is a header known as **planet_type.**
3. We find out the different unique values for planet_type. Planet type is the 7th element of the list in all the planet's data (We are not using the low gravity planets only since currently we are trying to find an insight).

```
planet_type_values = []
for planet_data in planet_data_rows:
 planet_type_values.append(planet_data[6])

print(list(set(planet_type_values)))
```

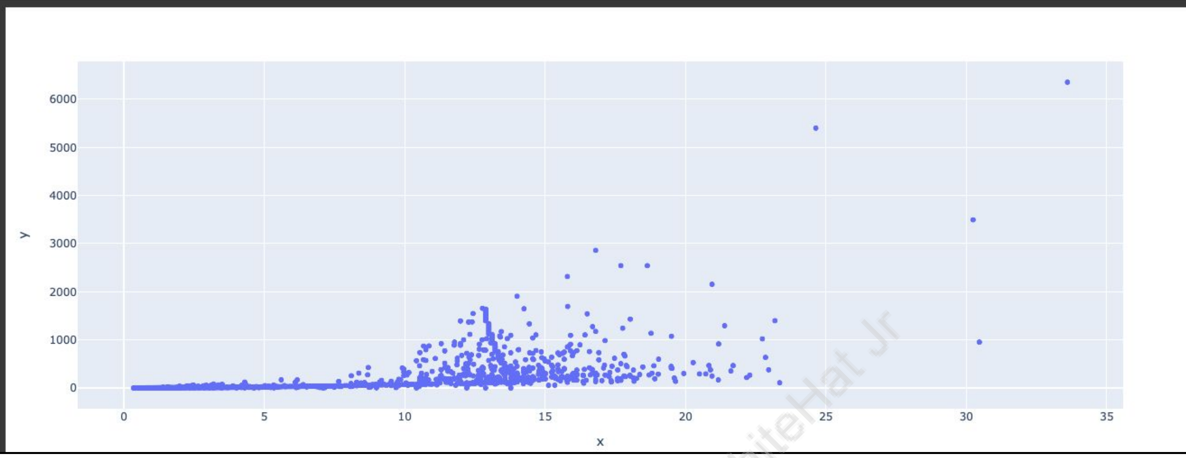
4. Here, we use the **set()** function to remove all the duplicates and keep the unique values only.
5. We get 4 different types of planets such as:
    - **Neptune-like - These planets are like Neptune! They are big in size and they are made of ice.**
    - **Super Earth - These are the planets that have mass greater than Earth but smaller than that of Neptune (Neptune is 17 times of Earth).**
    - **Terrestrial - It is a planet that is composed primarily of silicate rocks and metals (Earth, Mars).**
    - **Gas Giant - These are the planets that are mainly composed of Gases (Helium and Hydrogen) like Jupiter and Saturn.**
6. Let's plot a scatter plot where we have X-Coordinate as the radius of the Planet and Y-Coordinate as the mass of the planet.

```
planet_masses = []
planet_radiuses = []
for planet_data in planet_data_rows:
 planet_masses.append(planet_data[3])
 planet_radiuses.append(planet_data[7])

fig = px.scatter(x=planet_radiuses, y=planet_masses)
fig.show()
```

```
planet_masses = []
planet_radiuses = []
for planet_data in low_gravity_planets:
  planet_masses.append(planet_data[3])
  planet_radiuses.append(planet_data[7])

fig = px.scatter(x=planet_radiuses, y=planet_masses)
fig.show()
```



7. With the following results, we cannot detect the number of clusters it has just by looking at it. Let's find out the K of this data, which is the number of clusters it has.

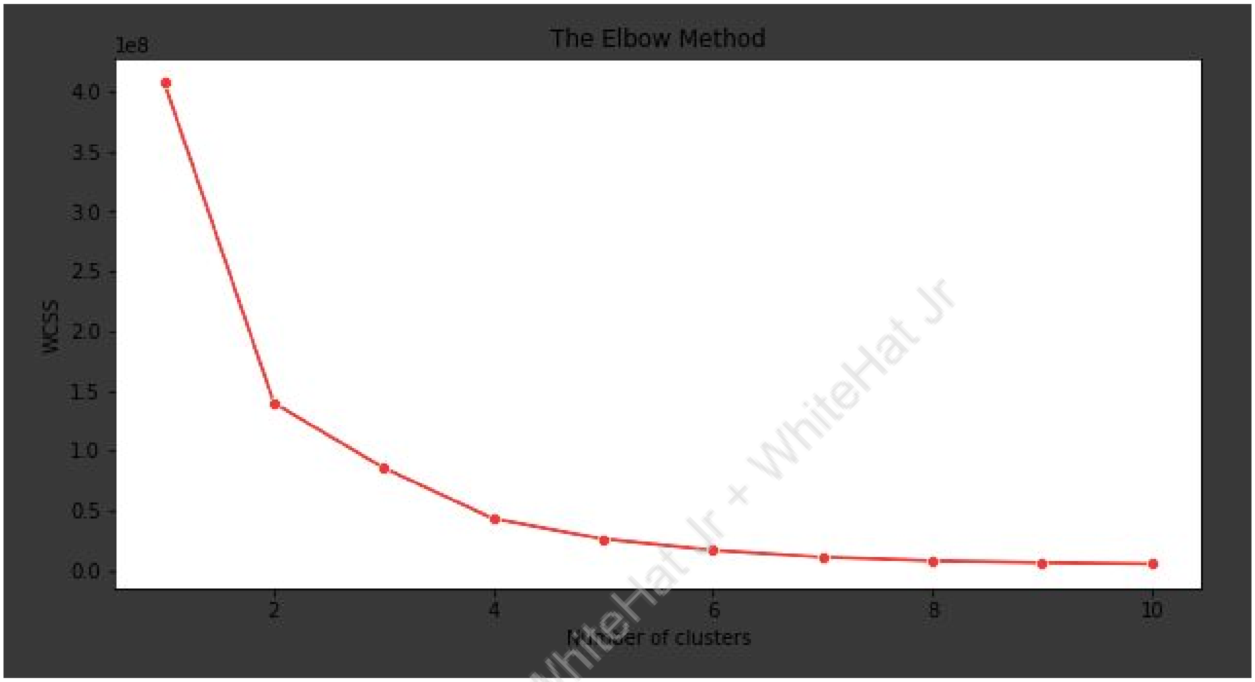
```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns

X = []
for index, planet_mass in enumerate(planet_masses):
 temp_list = [
               planet_radiuses[index],
               planet_mass
           ]
 X.append(temp_list)

wcss = []
for i in range(1, 11):
   kmeans = KMeans(n_clusters=i, init='k-means++', random_state = 42)
   kmeans.fit(X)
   # inertia method returns wcss for that model
   wcss.append(kmeans.inertia_)

plt.figure(figsize=(10,5))
sns.lineplot(range(1, 11), wcss, marker='o', color='red')
```

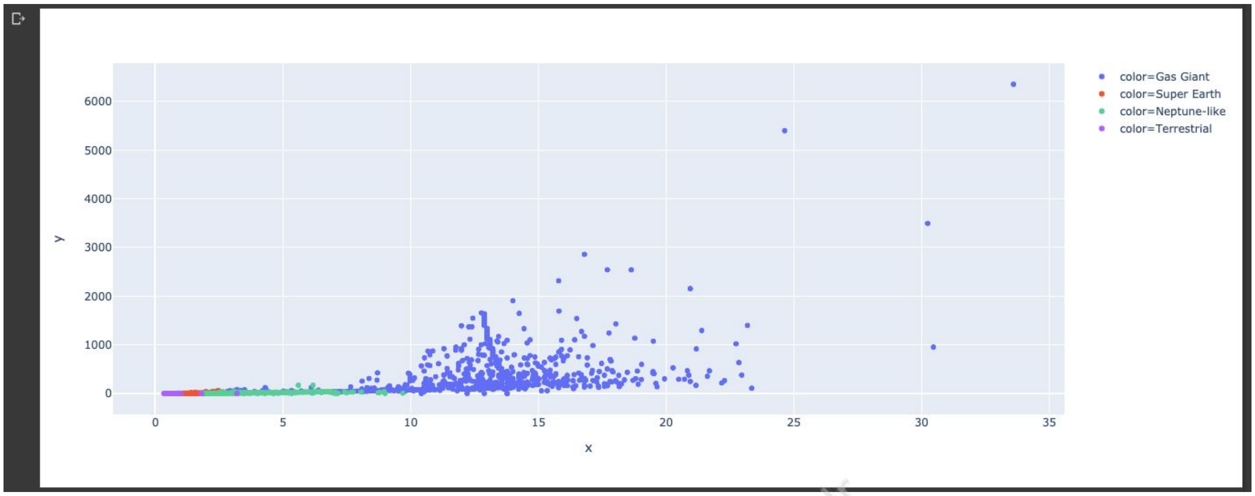
```
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



8. Here, using the **Elbow method** (remember when we learnt about K-Means and Clustering) we can see a descent up to **4**, after which it is not significant.
9. When we found the number of unique planet types earlier, there were 4 types but when we looked at the scatter plot, we couldn't detect the number of clear clusters. Machine learning can do things that we ourselves couldn't do!
10. Plot the previous chart with color coding based on the planet's type.

```
planet_masses = []
planet_radiuses = []
planet_types = []
for planet_data in low_gravity_planets:
 planet_masses.append(planet_data[3])
 planet_radiuses.append(planet_data[7])
 planet_types.append(planet_data[6])

fig = px.scatter(x=planet_radiuses, y=planet_masses, color=planet_types)
fig.show()
```



11. Given what we have learnt about different types of planets, we can say that life can be possible only on **terrestrial planets and super earth like planets.** At least, for us. Let's take the list of planets with low gravity and filter out more planets based on their type and create a new list.

```
suitable_planets = []
for planet_data in low_gravity_planets:
 if planet_data[6].lower() == "terrestrial" or planet_data[6].lower() == "super earth":
  suitable_planets.append(planet_data)

print(len(suitable_planets))
```

```
suitable_planets = []
for planet_data in low_gravity_planets:
  if planet_data[6].lower() == "terrestrial" or planet_data[6].lower() == "super earth":
    suitable_planets.append(planet_data)

print(len(suitable_planets))
```

```
1452
```

12. We are now left with 1,452 planets that can host us.

What's NEXT?

In the next class, We will try to filter out more planets based on various other factors and learn more about planets. We will also explore some new skills and concepts!

EXTEND YOUR KNOWLEDGE:

You can read the following blog on data processing to understand more:
https://www.zmescience.com/science/types-of-planets-feature/