Topic	DATA SCIENCE - 4	
Class Description	**The student will be comparing the habitable planets with Mars and would also be learning about more factors that make a planet habitable for us to survive.**	
Class	**PRO C134**	
Class time	**45 mins**	
Goal	• Merging the lists to get the final list of habitable planets • Comparing with Mars • Learning about more factors that make a planet habitable.	
Resources Required	• Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Smartphone • Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm-Up** **Teacher-Led Activity 1** **Student-Led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **05 mins**
Credit & Permissions:	**Exoplanet Exploration by NASA**	
WARM-UP SESSION - 10 mins		
Teacher Starts Slideshow **Slide # to #** 		

<table>
<tr><td colspan="2"><Note: Only Applicable for Classes with VA>
Refer to speaker notes and follow the instructions on each slide.</td></tr>
<tr><td>Teacher Action</td><td>Student Action</td></tr>
<tr><td>Hey <student's name>. How are you? It’s great to see you!
Are you excited to learn something new today?

In the last class, we learned about the speed of a planet!

Can you tell me its formula?

Following are the WARM-UP session deliverables:
● Greet the student.
● Revision of previous class activities.
● Quizzes.</td><td>ESR: Hi, thanks!
Yes, I am excited about it!

Click on the slide show tab and present the slides

ESR:
- The speed of a planet is the distance covered by it divided by the time taken by the planet to cover the distance.</td></tr>
<tr><td colspan="2">WARM-UP QUIZ
Click on In-Class Quiz</td></tr>
<tr><td colspan="2">Continue WARM-UP Session
Slide # to #
<Note: Only Applicable for Classes with VA></td></tr>
<tr><td colspan="2">Activity Details

Following are the session deliverables:
● Appreciate the student.
● Narrate the story by using hand gestures and voice modulation methods to bring more interest in students.</td></tr>
<tr><td>Teacher Action</td><td>Student Action</td></tr>
</table>

Awesome! In today's class, we will be finding out the final list of habitable planets (if any) and we will learn about planets in our solar system. We will also be learning about more factors that make a planet habitable. Isn't it interesting? Let's get started!	**ESR:** "Yes!"
Teacher Ends Slideshow 	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
ACTIVITY • **Review the concepts learned in the earlier classes**	
Teacher Action	**Student Action**
In the last class, we came up with 2 lists - **goldilock_planets** and **speed_supporting_planets**.	

```
goldilock_planets = list(suitable_planets)

temp_goldilock_planets = list(suitable_planets)
for planet_data in temp_goldilock_planets:
  if planet_data[8] < 0.38 or planet_data[8] > 2:
    goldilock_planets.remove(planet_data)

print(len(suitable_planets))
print(len(goldilock_planets))
```

```
696
25
```

```
planet_speeds = []
for planet_data in suitable_planets:
  distance = 2 * 3.14 * (planet_data[8] * 1.496e+8)
  time = planet_data[9] * 86400
  speed = distance / time
  planet_speeds.append(speed)

speed_supporting_planets = list(suitable_planets)

temp_speed_supporting_planets = list(suitable_planets)
for index, planet_data in enumerate(temp_speed_supporting_planets):
  if planet_speeds[index] > 200:
    speed_supporting_planets.remove(planet_data)

print(len(speed_supporting_planets))
```

```
676
```

Do you remember what **goldilock_planets** were?

ESR:
They were the planets that reside in the habitable zone of their solar system (0.38AU to 2AU).

Note: *Open Teacher Activity 1 and run all the cells of code. It is the continuation of the previous class.*

Awesome. Now, that we have the two lists, let's merge them to get the list of final planets that we could get based on the habitable data.

Let's start by creating a new list of **habitable_planets**. We are iterating over all the planets that support speed and we are checking if this planet is in goldilocks planets or not. If it is, we are appending this planet to our **habitable_planets** list.

```
habitable_planets = []
for planet in speed_supporting_planets:
  if planet in goldilock_planets:
    habitable_planets.append(planet)

print(len(habitable_planets))
```

```
25
```

We found 25 such habitable planets! Awesome!

But the question is, are these planets habitable? We don't know yet!

Let's see an example.

In our solar system, we know that the first 4 planets **Mercury, Venus, Earth,** and **Mars** are **Terrestrial Planets!**

Rest all the planets are either Gas Giants or Neptune-Like (and we do not have Super-Earth in our Solar System).

Thus, we can only look at our planets.
The first thing we filtered out is Gravity:

Mercury - 3.7m/s
Venus - 8.87m/s
Earth - 9.8m/s
Mars - 3.8m/s

This makes all these planets habitable.

Mercury - 0.4AU
Venus - 0.7AU
Earth - 1AU

Mars - 1.5AU

This again, makes all these planets lie in the Goldilock zone!

Now, if we talk about the speed of these planets, our final filter then:

Mercury - 47km/s
Venus - 35km/s
Earth - 30km/s
Mars - 24km/s

Again, this makes all these planets habitable!

Despite that, we are only considering the possibility to go to Mars and colonize it, instead of going to the other planets! This means that there are countless other reasons. Do you know a few of them?
Similarly, since we have these planets as terrestrial planets, we are still good to go!

Let's get into the reasons for our planets:

Mercury -
Mercury is not habitable since it does not have an atmosphere and its temperature varies from 100 Degree Celsius to 700 Degree Celsius.

Venus -
Venus is an extreme planet and it's very hot. Its atmosphere traps the heat on the surface and its temperature is a whopping 700 Degree Celsius. There are also rains of sulphuric acid.

Earth -

Seems like just the right planet to exist. **Mars -** Mars has some atmosphere and it also has water on its surface. The temperatures are a bit extreme (20 Degree to -150 degrees) but it is still manageable compared to others. The extreme temperatures are a result of the distance from the sun. Now, since we are saying that the Goldilock zone could be anywhere from 0.38 to 2 AU, and both Mercury and Venus lie in that zone, why are they so hot? These planets are hot because the Goldilocks zone is different for different stars. It depends on how big and powerful the star is, for us, we are at the beginning of our solar system's Goldilock zone and Mars is at the very end of it. To find exact goldilocks zones for different starts is a bit difficult and there's no such formula as yet, but the solar system having 0.38AU as a goldilocks zone would be much smaller than our sun and the solar system having a goldilocks zone higher than 1.5AU (could also be greater than 2, up to 10) are the stars that are much bigger and powerful than our sun!	**ESR:** Varied
Apart from this, we also have planets tidally locked to their stars. Do you know what that means? Tidally locked planets have the same period of rotation and revolution. As a result, the planet shows the same face toward its star. That's right. We have day and night cycles here, which means that we are not tidally locked.	**ESR:** Varied

You may have experienced that the days are warmer and the nights are cold! That's because we are facing toward the sun during the day and facing away from the sun during the night. When a planet is tidally locked, one side of the planet might be extremely hot while the other side of the planet might be extremely cold, which makes it impossible for us to exist. For reference, our moon is tidally locked to our planet Earth. We can always only see one side of the moon and the other side of the moon is never visible to us. We don't know what might lie there!	
Teacher Stops Screen Share	
So now it's your turn. Please share your screen with me.	
Teacher Starts Slideshow  **Slide # to #** <**Note**: Only Applicable for Classes with VA> Refer to speaker notes and follow the instructions on each slide.	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	
Teacher Ends Slideshow 	
STUDENT-LED ACTIVITY - 20 mins	
• **Ask the student to press the ESC key to come back to the panel.**	

- **Guide the student to start Screen Share.**
- **The teacher gets into Full Screen.**

Student Initiates Screen Share

ACTIVITY

- **Student code to segregate the lists of different planets!**

Teacher Action	Student Action
Now that we have performed our analysis and learned about a lot of concepts of science and space, it's time for us to start working on the app! We will be making a mobile app where we display all the planets and provide stats about the planet, why it's suitable, why it isn't, etc. ***Note:*** *Guide the student to open Student Activity 1 and check the filtered data.* For that, let's revisit all the steps we have done so far. Can you tell me about it? Great! Now let's reverse engineer our data and prepare 5 lists that contain all the data:	**ESR:** 1. We filtered out planets based on their gravity. 2. We then filtered out planets based on their type. 3. We then filtered out planets based on whether they are in the Goldilock zone or not and if their speeds support us, humans. 4. We prepared the final list of planets.

1. Planets that support only gravity. 2. Planets that support both gravities and are of suited types. 3. Planets that support all gravity, are of suitable type and fall in the goldilocks zone. 4. Planets that support all gravity, are of suitable types and have the suitable speed that we can survive.	
Planets that are potentially habitable support all the above filters. Now, you may say that we already have such lists, but if we just take the list of planets that support gravity (3,951) but this also contains the planets that are of suitable planet types in this. We only need those planets that just support gravity and nothing else. We need to ensure that the sum of all the planet lists comes out to be 4,250 (since we filtered out a few planets early on from here), which is the exact number of planets we started with. We need to iterate over all the planets and see what features they support. We can maintain a dictionary of all the features supported by a planet. 1. Create a dictionary called **final_dict**, iterating over our original **planet_data_rows** using **for** loop and using the **enumerate()** function. 2. Create a **features_list** where we want to keep track of all the features.	
Do you remember the formula we used to calculate the gravity of the earth? Great!! 3. So, calculate the gravity with the same formula.	**ESR:** $g = \frac{G * M_{earth}}{d^2}$

Note: *Guide the student to recollect and rewrite the formula.*

```
final_dict = {}

for index, planet_data in enumerate(planet_data_rows):
  features_list = []
  gravity = (float(planet_data[3])*5.972e+24) / (float(planet_data[7])*float(planet_data[7])*6371000*6371000) * 6.674e-11
```

We'll then try to apply all the filters we have done so far inside the **try** and **except** blocks!

The reason why we are using exception handling here is to handle any forms of errors that may come in comparing two values, etc.

4. Check the **gravity** inside **the try** block, if it's less than **100** then append **'gravity'** in the **feature_list** of the planet.

```
final_dict = {}

for index, planet_data in enumerate(planet_data_rows):
  features_list = []
  gravity = (float(planet_data[3])*5.972e+24) / (float(planet_data[7])*float(p
  try:
    if gravity < 100:
      features_list.append("gravity")
  except: pass
```

5. Apply all the filters for the Terrestrial planet and Super-Earth. Also for Goldilocks planet.
6. Calculate the speed and filter that planet out as well.

```
final_dict = {}

for index, planet_data in enumerate(planet_data_rows):
  features_list = []
  gravity = (float(planet_data[3])*5.972e+24) / (float(planet_data[7])*float(planet_data[7])
  try:
    if gravity < 100:
      features_list.append("gravity")
  except: pass
  try:
    if planet_data[6].lower() == "terrestrial" or planet_data[6].lower() == "super earth":
      features_list.append("planet_type")
  except: pass
  try:
    if planet_data[8] > 0.38 or planet_data[8] < 2:
      features_list.append("goldilock")
  except: pass
  try:
    distance = 2 * 3.14 * (planet_data[8] * 1.496e+8)
    time = planet_data[9] * 86400
    speed = distance / time
    if speed < 200:
      features_list.append("speed")
  except: pass
```

Finally, add this to the dictionary with the features as the value and the index of the planet as the key.

```
final_dict = {}

for index, planet_data in enumerate(planet_data_rows):
  features_list = []
  gravity = (float(planet_data[3])*5.972e+24) / (float(planet_data[7])*float(planet_data[7])
  try:
    if gravity < 100:
      features_list.append("gravity")
  except: pass
  try:
    if planet_data[6].lower() == "terrestrial" or planet_data[6].lower() == "super earth":
      features_list.append("planet_type")
  except: pass
  try:
    if planet_data[8] > 0.38 or planet_data[8] < 2:
      features_list.append("goldilock")
  except: pass
  try:
    distance = 2 * 3.14 * (planet_data[8] * 1.496e+8)
    time = planet_data[9] * 86400
    speed = distance / time
    if speed < 200:
      features_list.append("speed")
  except: pass
  final_dict[index] = features_list

print(final_dict)
```

Check the output now. The **final_dict** has all the features of all the planets.

```
27: ['gravity'], 28: ['gravity'], 29: ['gravity', 'planet_type', 'goldilock'], 30: ['gravity'],
```

Now, we have a dictionary that contains the index of all the **planet_data** as key and the values are the features of this planet that can help support life!

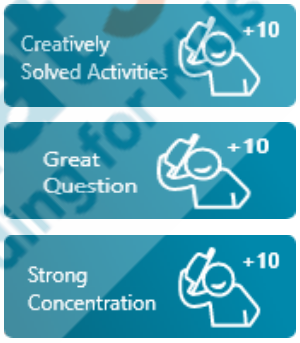
So, in this class, we merged and came up with a final list of features for all the planets that we can use for our Flask API.

We also learned about some other factors that are crucial in determining if the planet is habitable or not (with

reference from our solar system). How was your experience?	**ESR:** Varied
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 05 mins	
Teacher Starts Slideshow  **Slide # to #** <**Note**: Only Applicable for Classes with VA>	
Activity details **Following are the WRAP-UP session deliverables:** • Appreciate the student. • Revise the current class activities. • Discuss the quizzes.	
WRAP-UP QUIZ Click on In-Class Quiz	
Continue WRAP-UP Session  **Slide # to #** <**Note**: Only Applicable for Classes with VA>	
Activity Details **Following are the session deliverables:** • Explain the facts and trivia • Next class challenge • Project for the day • Additional Activity (Optional)	

FEEDBACK

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get “hats-off” for your excellent work! Amazing. While working on this project, we also made sure that we are at the top of all the concepts we have acquired so far. In the next class, we will cross-verify this dictionary's data with the data we found out and then create a Flask API for the same, so we can create a mobile app with a catalog of the analysis and data for all the exo-planets!	*Make sure you have given at least 2 hats-off during the class for:* 

PROJECT OVERVIEW DISCUSSION

Refer the document below in Activity Links Sections

Teacher Clicks ✖ End Class

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Boilerplate Code	https://colab.research.google.com/drive/1EottBCAunxmz2NrqEYm174assnJVxZg5?usp=sharing
Teacher Activity 2	Reference Code	https://colab.research.google.com/drive/14O6yFQfE7BTv2R176p31l_Dl5ttie9yl?usp=sharing
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/8bf91dbb-6f2b-4aab-aeb2-0b52c00e70ab.pdf
Teacher Reference 2	Project Solution	https://colab.research.google.com/drive/1kAQISnavPPeMuRRJLd9pmXBY-L4oqOGy?usp=sharing
Teacher Reference 3	Visual-Aid	Will be added after VA creation
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/6b8b4bf7-2a32-4587-b361-7f7236dc784e.pdf
Student Activity 1	Boilerplate Code	https://colab.research.google.com/drive/1FkEx_E-0Q1Wwotbeh2X5hYFn4sXf1S3r?usp=sharing