Ani Avetian, Bradley Knorr
May 28, 2020
CSE 163 Project Part 2

Educating the Public About What Lies Outside Our Solar System

Summary of Research Questions

1. **What is the average number of planets in each solar system? (Pandas & Graphing) Sub Question: What is the max amount of planets found for a solar system? (Pandas)**
   a. **Analysis:** For this question we are going to compute the mean number of planets in each solar system (group by + sum, then mean) using pandas. Our dataset contains “host names” which correspond to a solar system of exo-planets. Using the number of planets in each solar system we can use pandas again to calculate the mean of those numbers. To answer the sub question we will find the ‘hostname’ that contains the most planets using pandas and a max function. We will also use a boolean value to tell the function if it should plot a distribution plot on the number of planets vs. how many solar systems have that many planets. Using seaborn and matplotlib libraries we will be able to make our plot.
   b. **Why are we asking this?:**We are asking this question because we want to know how many planets are in the average solar system. Are we very large with 8 planets, or just an average solar system? We are asking the sub question because we want to know if there are solar systems that have more planets in their solar system than ours does.
   c. **Answer**: There is an average of 1.3469 planets in each solar system. The max number of planets for a solar system is 8.
2. **What is the mean mass of exoplanets in each solar system? (Pandas)**
   a. **Analysis:** For this question we are going to compute the mean mass of planets for each solar system (group by + mean) using pandas. Our dataset contains “host names” which correspond to a solar system of exo-planets. Using the masses from the planets and pandas the average mass can be computed.
   b. **Why are we asking this?:** We are asking this question because we want to know how massive planets are in certain solar systems.
   c. **Answer**: The average mass for each solar system varies a lot between each one. Some have a low mean mass and others have a high one (the mass is measured by Earth Masses).
3. **Which exoplanets are in the Habitable zone? (Pandas and Graphing)**
   a. **Analysis:** Using pandas and some math, we can test to see how many exo-planets found are habitable by being in the Goldilocks zone. We will plot the planets’ distance from the star (Semi-major Axis) vs the size of the star(Stellar Masses) to visually see which planets are in the habitable zone. Using seaborn and matplotlib, we will use a scatter plot to show the habitable planets vs the rest of

the dataset. We will also use those libraries to make a line plot for just the habitable planets and their distance from the star vs the size of their star. We have a lecture about how the habitable zone is calculated. We're hoping to use some variation of these formulas to calculate planets in the habitable zone:

   i. Some math on how to determine if the planet is in the habitable zone https://www.astro.umd.edu/~miller/teaching/astr380f09/lecture14.pdf
   ii. The goal will be to create a function that pandas can use to filter the dataset. The function will take certain values as parameters (such as temperature of star and semi-major axis) and then use those numbers to calculate if the planet is in the habitable zone, returning either a temperature estimate for the planet (0-100°C is habitable). Further explanation described in the Methodology section.

   b. **Why are we asking this?:** We are asking this because we know the goldilocks zone is a spot where water can exist in liquid form and life is possible. We want to know how many planets are actually in this zone. It is also a precursor to our next question.
   c. **Answer**: We saw that there were 160 exoplanets in the habitable zone. Using our graphs we were able to see a trend of where habitable planets lie in respect to every other planet.

4. **Are there exo-planets that could potentially hold life? (Pandas)**
   a. **Analysis:** Using the previous question's habitable zone calculation, we will further limit our scope on the dataset to see which of those planets have the features suitable for life. Features we will use beyond determining if the planet is in the habitable zone are planet density and orbit eccentricity to determine if planets may be habitable for life (we were hoping for more variables, but our dataset doesn't have atmospheric data). These two features are important for determining potential for life. Planet density determines if the planet is a gas planet or rocky planet. Eccentricity determines how elliptical a planet's orbit is; too elliptical and the temperatures will vary too wildly to support life very well. We will find the range of values for each feature that make a planet habitable for life.We will also use matplotlib and seaborn to show where the planets that have life are located (distance to their star vs size of star) compared to the habitable planets and the rest of the dataset. Further explanation described in the Methodology section.
   b. **Why are we asking this?:** We are asking this question because we want to know if there is potential life outside Earth and how many planets meet that criteria.
   c. Answer: There are 2 exoplanets that could potentially hold life.

5. **How accurately can you predict the mass of a planet? (ML)**
    a. **<u>Analysis:</u>** (Note: all ML methodologies are almost identical) We are going to create a machine learning *regression* model to predict the mass of an exoplanet based on other known information about them. Then test the accuracy of that model. The accuracy will be defined by the mean absolute error, r-squared, and visually comparing plots. For this question we are assuming we know nothing about the mass of the planet. The steps for this model are to separate the features and label column, divide the data into training and test data, use the training data to create a model, and use the test data to test the accuracy of the model. The plot we used was a distribution plot (usually known as a histogram) to compare the results. It's a subplot with 2 plots: the top for the predicted values and the bottom for the actual values. X axis is the label, the y axis is the number of planets in that range of the bar. We used searborn to put the data on the plot, and matplotlib to change the axes and add a title.
    b. Labels: Mass of planet, features: number of planets in the solar system, planet's orbital period, orbital eccentricity, planet's distance from its star, planet's radius, planet's density, the star's temperature, the star's mass, and the star's radius
    c. **<u>Answer</u>**: Machine learning is a decent way to predict the mass of a planet, but not a great way.

6. **How accurately can you predict the distance from the star? (ML)**
    a. **<u>Analysis:</u>** (Note: all ML methodologies are almost identical) We are going to create a machine learning *regression* model to predict the distance of the exoplanet from its parent star based on other known information about them. Then test the accuracy of that model. The accuracy will be defined by the mean absolute error, r-squared, and visually comparing plots. For this question we are assuming we know nothing about the planets' distance from the star. The steps for this model are to separate the features and label column, divide the data into training and test data, use the training data to create a model, and use the test data to test the accuracy of the model. The plot we used was a distribution plot (usually known as a histogram) to compare the results. It's a subplot with 2 plots: the top for the predicted values and the bottom for the actual values. X axis is the label, the y axis is the number of planets in that range of the bar. We used seaborn to put the data on the plot, and matplotlib to change the axes and add a title.
    b. Labels: Distance from star (semi-major axis), features: number of planets in the solar system, planet's orbital period, orbital eccentricity, planet's radius, planet's mass, planet's density, the star's temperature, the star's mass, and the star's radius
    c. **<u>Answer</u>**: Machine learning appears to be a good way to predict a planet's distance from a star.

7. **How accurately can you predict the eccentricity of an exoplanet's orbit? (ML)**
    a. **Analysis:** (Note: all ML methodologies are almost identical) We are going to create a machine learning *regression* model to predict the eccentricity of the exoplanet's orbit based on other known information about them. Then test the accuracy of that model. The accuracy will be defined by the mean absolute error, r-squared, and visually comparing plots. For this question we are assuming we know nothing about planets' orbits aside from distance from the star (semi-major axis). The steps for this model are to separate the features and label column, divide the data into training and test data, use the training data to create a model, and use the test data to test the accuracy of the model. The plot we used was a distribution plot (usually known as a histogram) to compare the results. It's a subplot with 2 plots: the top for the predicted values and the bottom for the actual values. X axis is the label, the y axis is the number of planets in that range of the bar. We used seaborn to put the data on the plot, and matplotlib to change the axes and add a title.
    b. Labels: eccentricity of orbit for a planet, features: number of planets in the solar system, planet's orbital period, planet's distance from its star, planet's radius, planet's mass, planet's density, the star's temperature, the star's mass, and the star's radius
    c. **Answer**: Machine learning is not a very effective way at predicting the eccentricity of a planet's orbit

Why are we asking this?: For questions 5-7 we are researching these questions using machine learning because there are a lot of unknowns for exo-planets and we want to see if using a machine learning model is a good way to predict the information we don't know. We will split the data into training (80%) and test (20%) data to determine the accuracy of the trained machine learning model.

8. **What is the distribution of the mass of the exoplanets? (graphing)**
    a. **Analysis:** Using our dataset we can look at the masses of the exo-planets and see what the distribution looks like for all the masses. We will then graph this data using the correct visualizations. Using matplotlib and seaborn we will plot a distribution plot to show the mass category and how many planets fall into that range of mass. We'll determine if we need to use log-scale axes based on the distribution of the data.
    b. **Why are we asking this?:** We are asking this question because we want to see how diverse the distribution of mass is for planets we have found. Using a plot, with the right encodings, we can convey the masses clearly.

c. **Answer**: Our graph showed us that a majority of masses were on the smaller side.

9. **What is the distribution of the distance from the star for the exoplanets? (graphing)**
    a. **Analysis:** We want to visually see the distribution of exoplanets and their distances from their parent star. To do this, we will plot the planets' distances (semi-major axis) from the star on a distribution plot. The libraries we will use to accomplish this are seaborn and matplotlib. We'll determine if we need to use log-scale axes based on the distribution of the data.
    b. **Why are we asking this?:** We are asking this because we want to know how far these exo-planets actually are from their star. Again, by using a plot, we can use the right encodings to convey the distances clearly.
    c. **Answer**: Our graph showed us that a majority of distances to planets were smaller than 1 AU or were about 1 AU.

Motivation and Background

Motivation: We want to gather information about exo-planets to inform the public about what lies just outside our solar system. In order to do this, we will use a range of questions to explain the features about these exo-planets. These questions are worth computing because we want people to know there is so much more than what is in our solar system and more is being created in our ever expanding universe. It would make a difference to know the answers because then people will understand that there are so many other planets and solar systems similar or different to our own. They will also understand how important it is to study these things because we could potentially find life elsewhere.

Dataset

DataSet From NASA on Confirmed Exo-Planets:
https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=planets

Documentation for the Dataset:
https://exoplanetarchive.ipac.caltech.edu/docs/API_exoplanet_columns.html

This dataset contains more than 4000 rows of confirmed Exo-Planet data. It contains information about where the exo-planet is, its eccentricity, distance from its star, how it was discovered, and many more features. The data comes from NASA's archive so we know that the data collected is reliable. The data is collected from observatories here on earth and telescopes like Kepler that are in space.

Note: The dataset we used contained more columns that we made sure to select and add in the data before we downloaded it. Using the 'README.md' document, there are clear instructions on how to select and add these rows before downloading the dataset.

Methodology

Analysis: For each question, point 'a' describes the analysis we will perform.
Challenge Goals: The questions that we are asking will directly connect with our challenge goals. Questions 5-7 relate to our 'machine learning' challenge goal because we are using machine learning to predict values. Although, all of the questions relate to 'many perspectives' because there are different paths we are taking to answer the questions. We want to inform the public with general information about exo-planets, provide a way to predict values about exo-planets, and we want to use specific calculations to find life outside Earth.

Questions 3 and 4 had much more advanced methodology and were not adequately explained in the *summary of research questions* section above, so we added to our methodology explaining here how we came up with our calculations and how they were used:

Question 3 Methodology:

To answer question 3, we had to create a formula to determine if a planet was in the habitable zone. To do that, first we had to define what it means to be in the "habitable zone". A general definition that scientists have come up with is a planet is in the habitable zone when it has a surface temperature that allows for liquid water, which is 0-100°C. Our dataset does not contain atmospheric data for these planets, so we can only take into account the effective temperature of a planet. If we did have that data, Earth, for example, may be warmer than it normally would be because of its thick atmosphere that traps heat. This effective temperature is defined as the temperature of the solar energy that the planet would be absorbing. If a planet is farther from a star, this effective temperature is lower, and if a planet is closer to a star it is higher. However, not all stars are the same. They can be bigger and burn brighter and hotter. The habitable zone for a tiny red dwarf star is much different than that of a supergiant star. So our goal was to come up with a formula to determine the effective surface temperature of a planet based on 3 factors: the radius (R) of the star (measured in solar radii), the effective surface temperature (T_s) of the star (measured in Kelvin), and the planet's distance (r) from the star (measured in AU as the Semi-Major Axis of the planet's orbit). To calculate this temperature took some advanced techniques we learned in our college physics classes. So if you're a fluent physics student, you can follow along with the process we took to come up with this formula:

It turned out to be much more complicated than we originally anticipated. There was no website or a simple formula to predict this temperature. We ended up combining 2 formulas from two different websites. The first formula was the general equation for calculating the temperature of a patch of space a certain distance from a star: [$\sigma T_p^4 = F = L/(4\pi r^2)$] which we got from [https://www.astro.umd.edu/~miller/teaching/astr380f09/lecture14.pdf]. The issue with this formula was it needed the luminosity (L) of the star, which we didn't have. This meant we needed to go to another website to get the formula for a star's luminosity. We found this formula to be: [$L = \sigma T_s^4 * (4\pi r^2)$] which we got from [https://astro.unl.edu/naap/hr/hr_background2.html]. Now that we were able to calculate the luminosity, we were able to plug in all the pieces to the formula. The two sides of the equation are connected using the total energy flux (F). We also had to get Stefan Boltzmann's constant (σ) which is [5.670367 x 10^{-8}]. With this information, we just had to solve the equation for the effective temperature of the planet (T_p). We end up with this equation: [$T_p = \sqrt[4]{((L/(4\pi r^2))/\sigma)}$]. Now that the entire formula is determined, we just run the numbers: use the star's radius (R) and effective temperature (T_s) to determine the star's luminosity, then use the luminosity (L) and planet's distance from the star (r) to calculate the planet's effective temperature (T_p). We ran into an issue with this after our first attempt: we found the earth's effective temperature to be 200,000°C. We knew this was wrong, but our formula was right, so we decided to look into the units of the variables in this equation. After we ran a dimensional analysis of the equation, we found the issue. The units for Stefan Boltzmann's constant (σ) are [$W * m^{-2} * K^{-4}$], and we were using solar radii and AU instead of meters. So we had to convert those variables before plugging them into the equation (1 au = 1.5×10^{11}m and 1 solar radii = 6.9634×10^{8} m). Once we converted those units to meters, the units canceled out in the equation as they should:

- $\sigma T_p^4 = F = L/(4\pi r^2)$
- $(W * m^{-2} * K^{-4}) * K^4 = W * m^{-2} = ((W * m^{-2} * K^{-4}) * K^4 * m^2) * m^{-2}$

Now our formula was giving us much more reasonable temperature estimates. The NASA dataset did have effective temperatures for some planets, but not all that many. There weren't enough temperature values to use it effectively for our habitability analysis, but we had a baseline to judge our calculations on. Obviously we trust NASA's calculations much more than ours. All of our numbers appeared to be a certain fraction off from NASA's numbers, so we decided to arbitrarily multiply our answers by just under ⅔ (200/308). Once we did this, our numbers were anywhere between 0.5-5% of theirs, often within a few degrees (excluding an outlier)! It became more inaccurate as temperatures got far below 0 or scorching hot, but we cared about the accuracy of planet temperatures in or around the habitable zone (0-100°C). All of these temperatures were accurate to a couple of degrees of NASA's, so we called our function a success and was able to be used to calculate planets in the habitable zone.

With a working planet temperature function, we just needed to use pandas to go through all the planets in the dataset and filter the ones within 0-100°C. We visualized this by plotting the planet's distance from the star vs. the size of the star and coloring the planets that were in the habitable zone blue, and all the rest of them red to show the distinction and allow us to look for a trend for planets in the habitable zone.

Question 4 Methodology:

Thankfully, this function was much easier than the previous one. To find the features of a habitable planet, we found a concise Wikipedia page (https://en.wikipedia.org/wiki/Planetary_habitability). The features we found were planet temperature (which we just calculated above), planet density (to determine if it's a rocky or gas planet), planet radius, planet mass, and orbital mechanics (we have eccentricity). We didn't have any atmospheric data, so these are the numbers we were able to use. Here are the range of values for each feature we found:

- **Temperature**: 0-100°C, the temperature that water is in its liquid form
- **Planet density**. All the rocky planets we looked at are above a density of 3 kg/m^3 and gas planets are less than 2, so we split the difference and arbitrarily said >2.5 is a rocky planet, and <2.5 is a gas planet
- **Planet radius**: The planet must have a radius >0.5x and <2.5x earth's radius
- **Planet mass**: The mass of the planet must be >0.3x and <10x earth's mass
- **Planet orbit eccentricity**: Planets that have a high eccentricity in their orbit have temperatures that vary too much to support life. Earth's orbit, for example, has an eccentricity of just 0.02. There was no definite eccentricity range that we could find, so we just arbitrarily said that any planet with an orbital eccentricity of <0.25 was habitable. This may be a bit generous, but we were hoping to have a fair amount of planets that could be considered suitable for life.

If the planet passed all 5 of these tests, we deemed the planet habitable for life. Once we had the planets habitable for life, we did some research on them to find out how accurate our model was and provided some information on these planets. We also showed them on the scatter plot we made for question 3.

Results

Note: AU is a pretty big measurement. Throughout our project, we saw this measurement. When we mention a smaller distance of exoplanets to their stars (and the units used are AU), it doesn't mean they are a few kilometers away. If they were, they would have been vaporized due to the temperature of their star. Below are a few conversions to show how large AU values are:

0.1 AU = $1.496 \times 10^{7} km$

0.5 AU = $7.48 \times 10^{7} km$

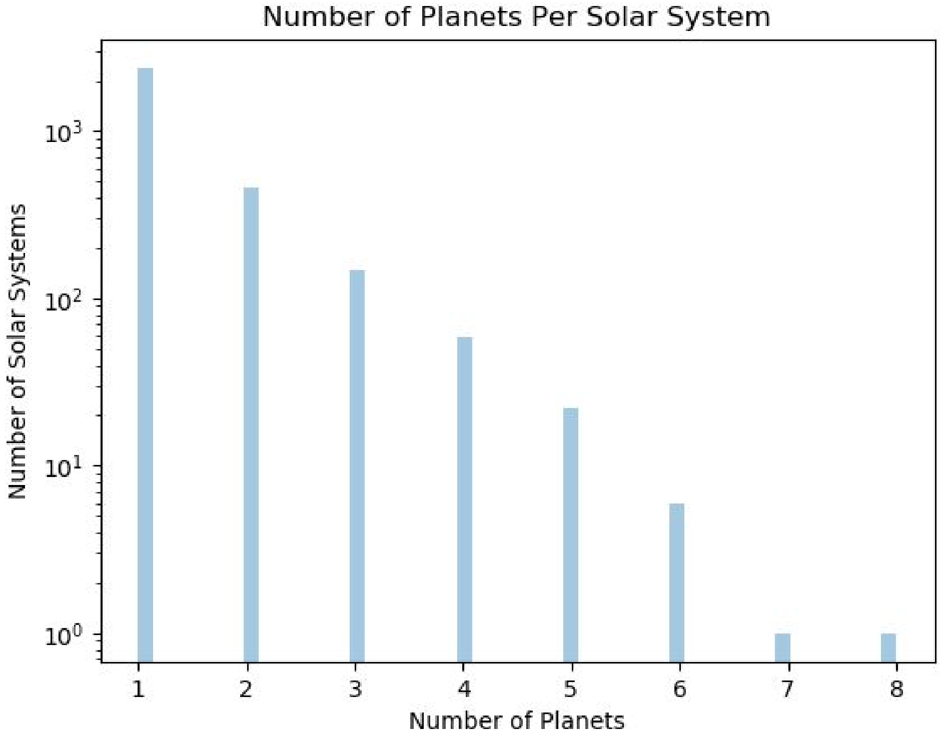
1 AU (Distance from our Sun to Earth): $1.496 \times 10^{8} km$

5.2 AU (Distance from our Sun to Jupiter): $7.7796 \times 10^{8} km$

Along with that, we use other astronomical measurements in our project. To measure the mass of planets, we use *earth masses* (5.97224×10^{24} kg). To measure the mass of stars, we use *stellar masses* (2×10^{30} kg). And to measure the radius of a star, we use *stellar radii* (696,000 km).

1. **What is the average number of planets in each solar system? (Pandas)**
   Sub Question: What is the max amount of planets found for a solar system? (Pandas)

   We saw that there was an average of 1.3469 planets in each solar system. This was surprising to see because we had more than 4000 rows of data. Although, this was expected. When going through the CSV file for the data, you can see that the majority of solar systems only have 1 or 2 planets. Rarely would you find a solar system with more than 6 exoplanets discovered. For our sub-question, the max amount of exoplanets found for a solar system was 8. Again, this was surprising because this is the number of planets we have in our solar system. We can trust this data because we used a smaller dataset to test and make sure the values were coming out correct. We manually did the math and compared it to the answers we got which all matched. The plot below visually shows you the answers we found. In the distribution plot, you can see that the majority of solar systems have 1, 2, or even 3 planets while 7 and 8 planets per system are not as common.



2. **What is the mean mass of exoplanets in each solar system? (Pandas)**

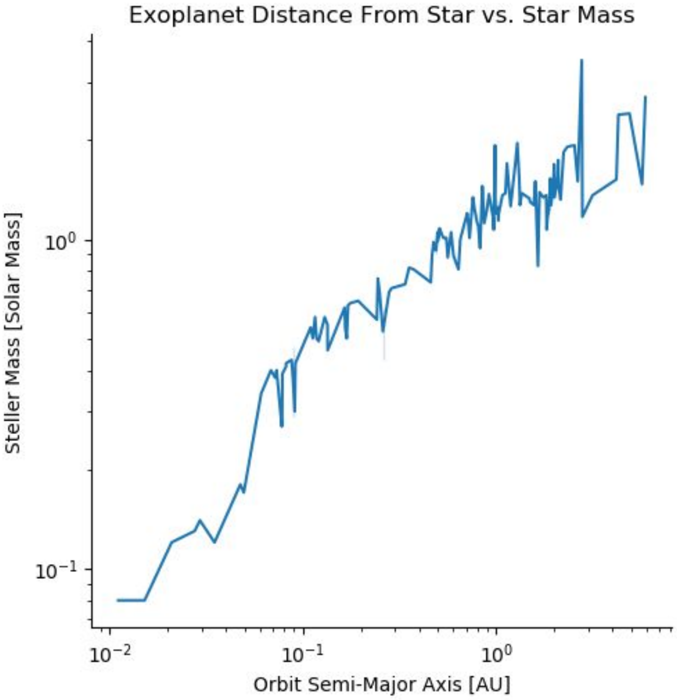
The range for the mean mass of exoplanets in each solar system varied a lot. Below is a table we created that contains the planet hostname on the left and the mean mass on the right. The table does not include all the solar systems because there are 759 solar systems in total. Although it includes some of the average masses for some solar systems.

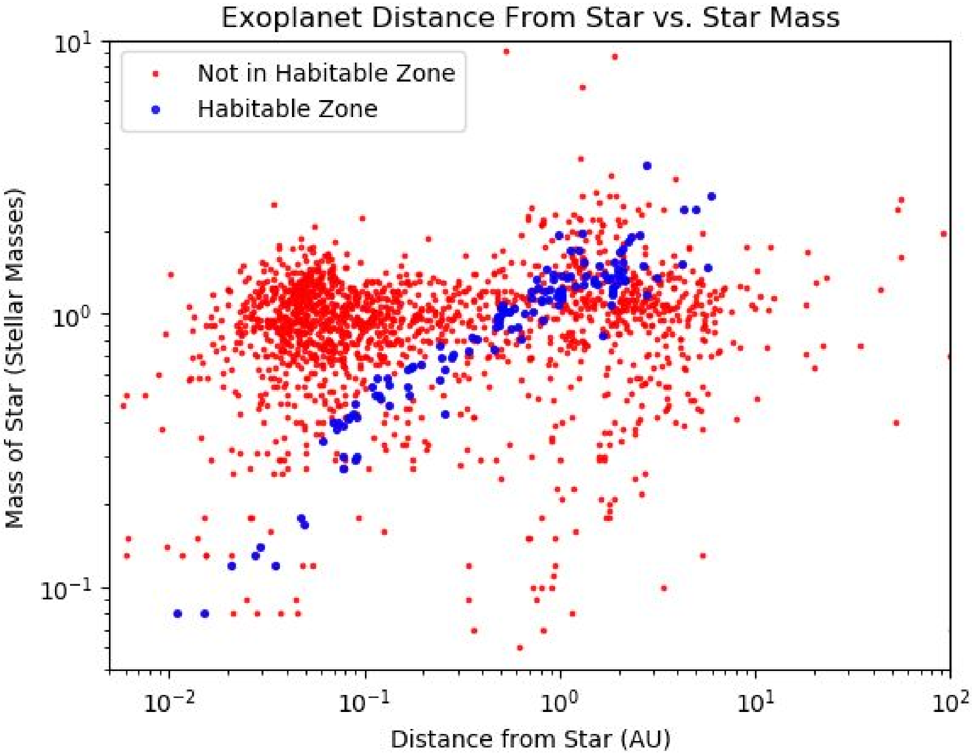
Planet Hostname	**Mean Mass [Earth Masses]**
1RXS J160929.1-210524	3000
2MASS J01225093-2439505	7786.4
2MASS J02192210-3925225	4417.837
2MASS J04414489+2301513	2383
2MASS J12073346-3932539	1271
bet Pic	2860
eps Eri	492
eps Ind A	1032
kap And	4327
nu Oph	7448

A majority of solar systems only had 1 planet and therefore this data reflects that one planet's mass. Although it was common to see 2 planets in a solar system so some of the values reflect the average mass of a couple of planets. The reason we made this table is because it was a cleaner way of representing the data frame we got back from our function that computed the mean masses.We can trust this data because we manually did it on a subset of the data. The values that came out manually matched the ones in the dataset.

3. Which exoplanets are in the Habitable zone? (Pandas and Graphing)

Using the calculations described above we saw that there were 160 exoplanets in the habitable zone. Even though we had a lot of data, we expected this number to be lower because we thought that there could not be so many planets in the habitable zone. Although there were. The line plot to the right shows a logarithmic model of the semi-major axis (distance to star) measured in AU to the stellar mass of the star measured in stellar masses. For the habitable planets, the farther you got away from the star the bigger the star became.

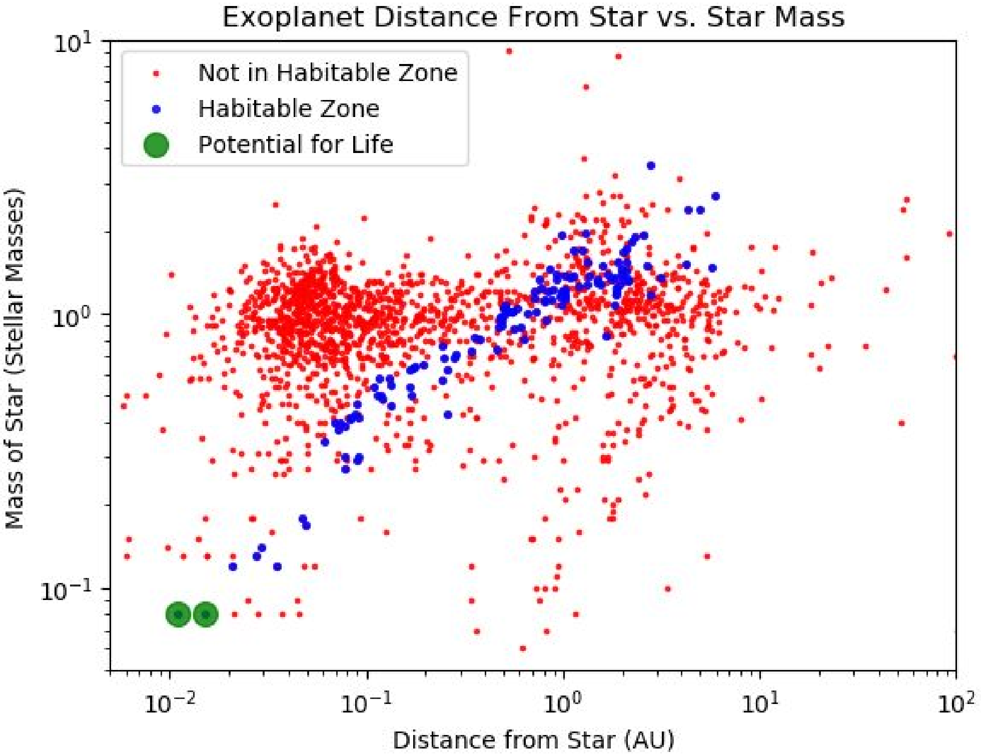




The plot above shows a scatter plot of distance from a star, measured in AU, to mass of the star, measured in stellar masses. The scatter plot shows data from two different datasets. The red points show all the data on exoplanets. We can see that a majority of these exoplanets are closer to their stars or are not in the habitable zone. The habitable planets (shown in blue) are more in the center of the scatter plot and follow a clear trend in this sub-dataset. This shows a direct correlation between star mass and a planet's distance from its star for it to be considered in the habitable zone. Since both axes are logarithmic, the relationship is exponential. The correlation makes sense because as the mass of the star becomes bigger it becomes hotter, and therefore moves the habitable zone farther away to keep the planet at the same temperature. The exponential correlation is expected because heat is distributed in three dimensions (4/3 πr^3), which means energy is lost exponentially across distance.

4. **Are there exo-planets that could potentially hold life? (Pandas and Graphing)**

Our calculations showed that out of all the planets in the habitable zone, only 2 could potentially hold life. The names of these planets are TRAPPIST-1 b and TRAPPIST-1 c. They met every single piece of criteria we set for life (outlined in Question 4 Methodology). In the scatter plot below we can see where those planets are relative to the rest of the data set (red) and the exoplanets in the habitable zone(blue). They are close to their star and their star is on the smaller side as well.

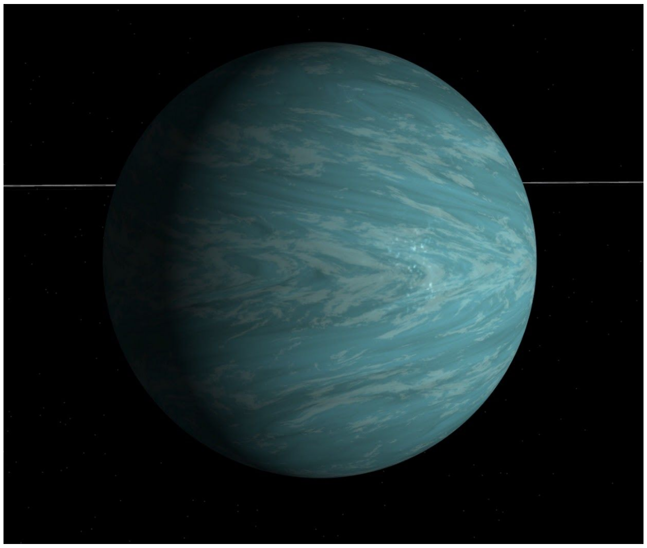


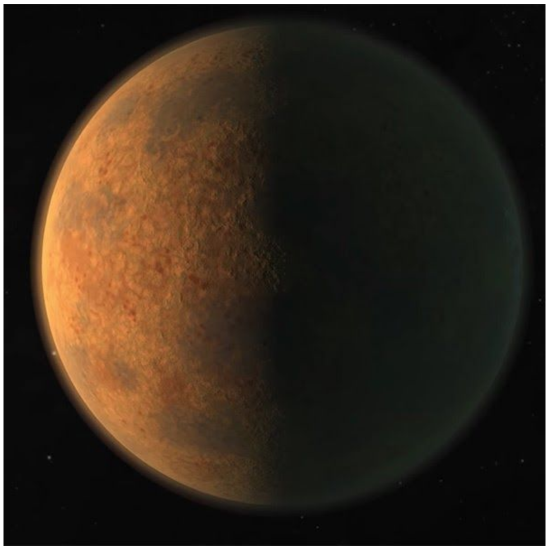
When doing further research on these planets using NASA's website, we discovered that these two planets are not in the habitable zone for their system. This means that our model for predicting which exoplanets are in the habitable zone has some flaws. NASA does provide effective temperatures for these planets: 126.5°C and 68.5°C respectively. So our model was off by at 28°C for TRAPPIST-1 b (98.715678°C) to be labeled in the habitable zone. However, TRAPPIST-1 c was within the *effective* temperature range for liquid water according to NASA, so clearly NASA has a more refined way to define the habitable zone than us. However, we didn't have the information or experts that NASA does, so this is the best we could do for a 4-week project. Despite this, the model was on the right track and did come knocking on the door to habitable planets. The entire TRAPPIST-1 system has been a goldmine for astronomers. Several of the planets appear to be rocky with similar features as the earth (mass, radius, density, etc). TRAPPIST-1 e, f, and g have all been labeled as potentially habitable (d is thought to be too close, but not confirmed) as we continue to research them. TRAPPIST-1 b and c have been found to be Venus-like with thick atmospheres and very hot surfaces based on transmission spectrums taken from them. With that caveat out of the way, here were some interesting facts we learned about the two planets our model found:

TRAPPIST-1 b is said to be a "potentially rocky world, larger than earth". The exoplanet is 0.011 AU away from its star and even if we traveled the speed of light (670 million miles per hour), it would take us 41 years to get there.

TRAPPIST-1 c is said to be a "potentially rocky world, larger than earth" just like TRAPPIST-1 b. Although, this exoplanet is farther from its sun at 0.015 AU away.

Below are artist renditions of what these two planets would look like. The one on the left is TRAPPIST-1 b and the one on the right is TRAPPIST-1 c.

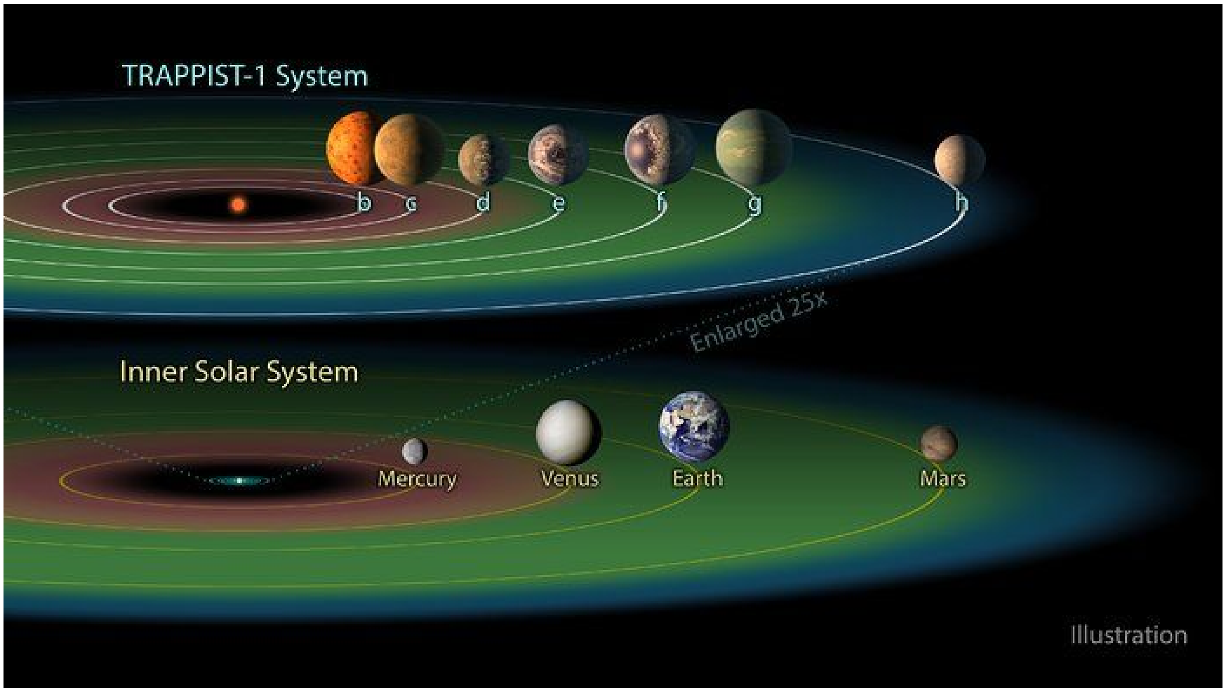




Information about the exoplanets from website.

Ani Avetian, Bradley Knorr
May 28, 2020
CSE 163 Project Part 2

Below is an image comparing our solar system to TRAPPIST-1's. The red area is assumed to create a runaway greenhouse effect, green is the habitable zone, and blue is thought to be too cold.

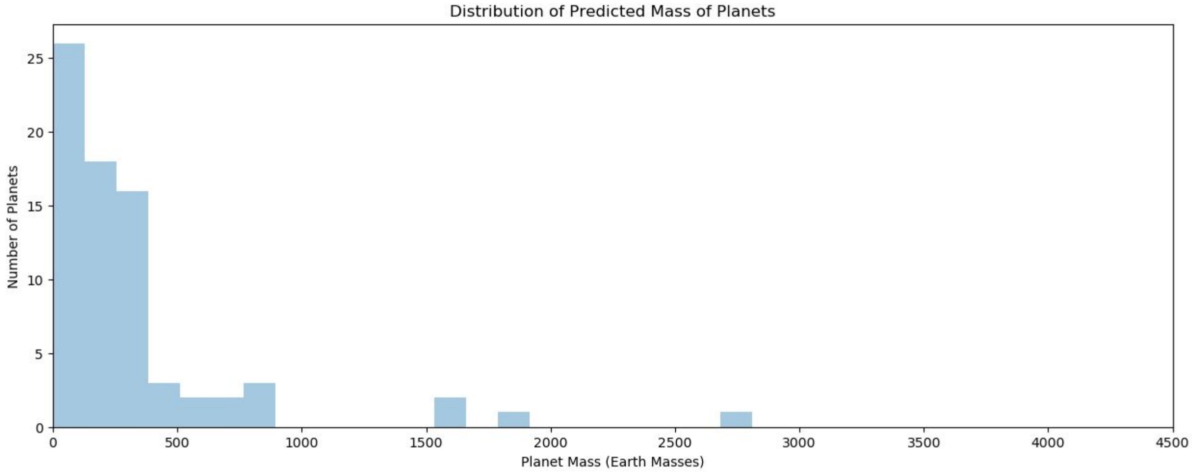


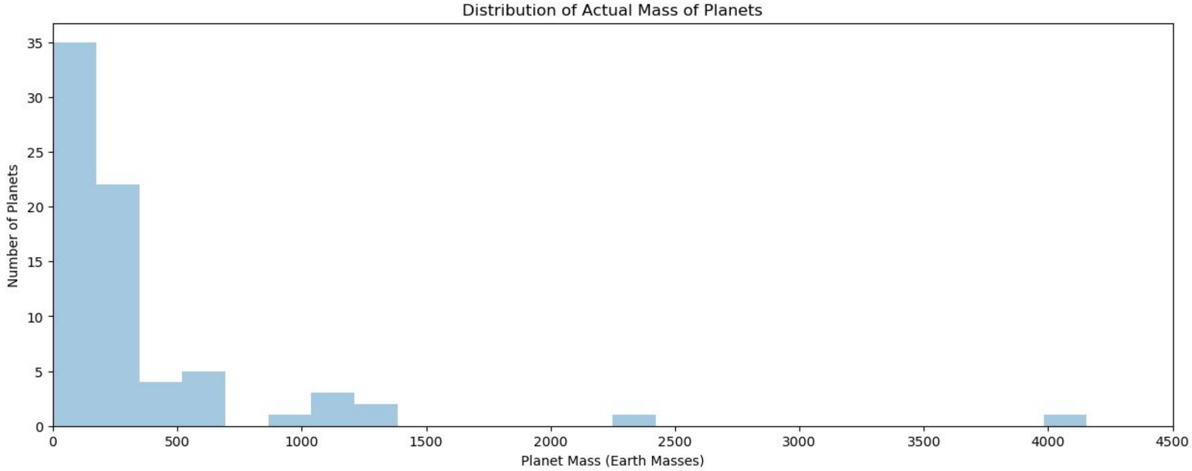
https://www.spacetelescope.org/images/heic1802d/

The most habitable or Earth-like planet found to date is Kepler-452b and is thought of as "Earth's cousin or Earth 2.0". We found this planet in our dataset and found that it was missing information such as planet mass and orbital eccentricity, so it never got to be considered for habitability beyond if it was in the habitable zone. We calculated its effective temperature to be -10.925000 °C, 11 °C short of what we defined as habitable, while Wikipedia gives it a −8 °C equilibrium temperature. When we saw that it has a predicted surface temperature of 120 °C, we realized that surface temperature and effective/equilibrium temperature could be very different; also that surface temperature goes well beyond the scope of our abilities because it involves examining its atmosphere, predicting greenhouse effects, determining the composition of the surface, and analyzing electromagnetic transmission spectrums. Due to these factors, the effective temperature tends to underestimate the true surface temperature of a planet. That explains why we found Kepler-452b (-10.925 °C) to be too cold for liquid water, as well as why we labeled TRAPPIST-1b (98.715678°C) and c (44.668021°C) as in the habitable zone rather than d (-5.460871°C), e (-39.616224°C), f, and g. We are still excited to have come up with a formula that allowed us to determine the temperature of a planet to the best of our abilities.

5. **How accurately can you predict the mass of a planet? (ML)**

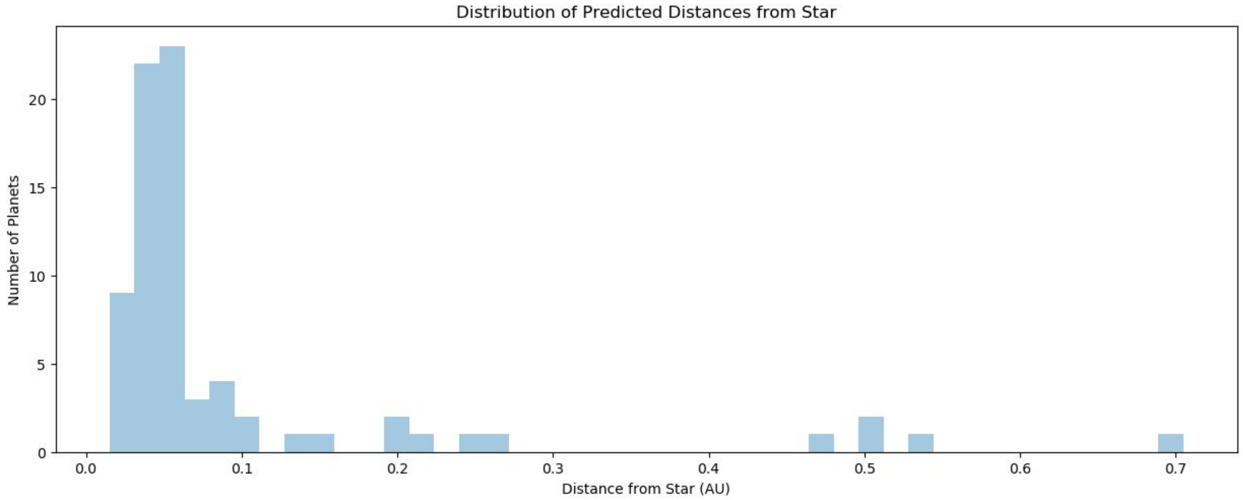
For this question, we were trying to see how accurate a machine learning model could be at predicting the mass of planets given certain information. Here were the features we provided: number of planets in the solar system, planet's orbital period, planet's distance from the star, planet's eccentricity, planet's radius, planet's density, the star's temperature, the star's mass, and the star's radius. Overall we would say that the machine learning model was somewhat effective at predicting the mass of planets. Based on the plots comparing the predicted masses to the actual masses, you can see that the shape of the plots are similar. You can dive deeper into the numbers of this model too. The mass calculations had an average accuracy to 151.123493 Earth masses, and the r-squared value was 0.845332. This means that on average, the estimations were off by 150 earth masses, which may not seem good. However, this number is skewed right because of the planets with huge masses and them being pretty far off. This did not seem like a good way to define accuracy, so we tried using the r-squared value. This gives a little better indication of how accurate the model was because we can see the distribution of the values, which compares each planet against itself (prediction vs actual) rather than against the whole dataset. An r-squared value of 1 is perfect, so 0.85 is pretty good, however this value is somewhat hard to define given our dataset. Combining all this information, we'd say that machine learning to predict mass of planets is not ideal, and should only be used if you're not looking for overly accurate results. However, for general use, this model does a good job predicting masses if the purpose is to fill holes in your dataset (as long as it's clear they aren't the true masses).

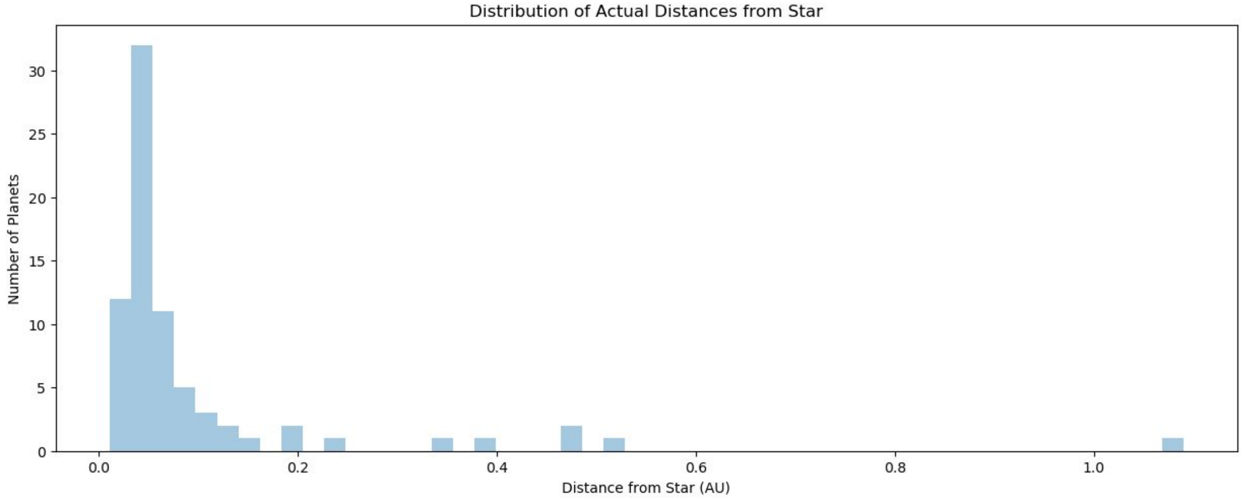




6. **How accurately can you predict the distance from the star? (ML)**

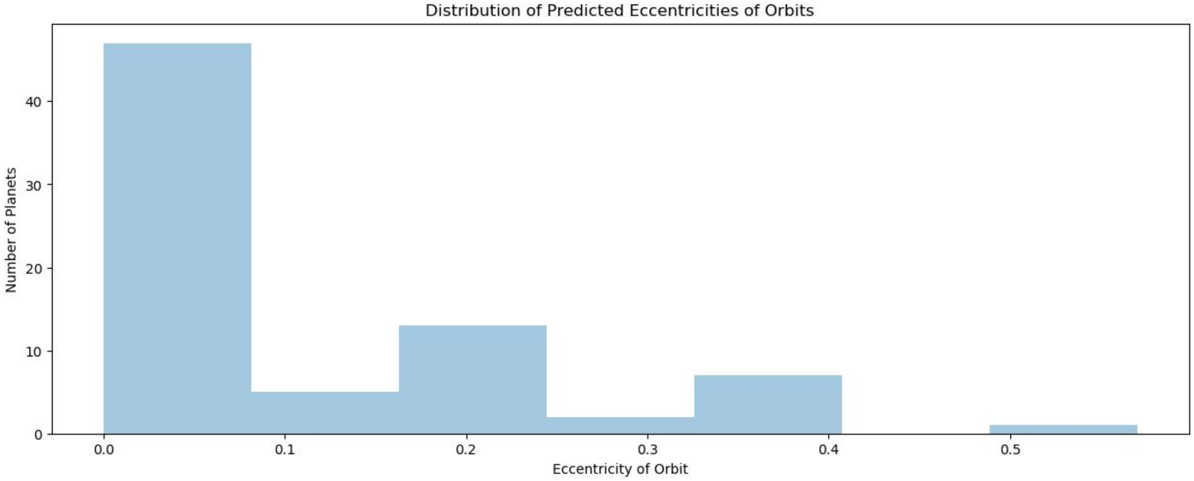
For this question, we were trying to see how accurate a machine learning model could be at predicting the planets' distances from their star given certain information. Here were the features we provided: number of planets in the solar system, planet's orbital period, planet's eccentricity, planet's radius, planet's mass, planet's density, the star's temperature, the star's mass, and the star's radius. Overall we would say that the machine learning model was effective in predicting the distance from its host star. Based on the plots below, you can see how accurate the model was. The overall shape of the graphs of the predicted vs actual values is very similar. Also, the values as the distance get larger are still fairly consistent. The statistics reflect this too. The distance calculations had an average accuracy to 0.014119 AU, and the r-squared value was 0.898887. Even at the scale of 0 to 1, 0.01 isn't all that much. With all the planet distances that this model needed to predict, being just over 1% off on average is very good, and is statistically significant (5%). The r-squared value of just short of 0.9 is also very good. A perfect r-squared value is 1, which means the values fit the curve 100%. In this case, “fitting the curve” means how well the predicted values aligned with the actual values. The ability to predict distances makes sense too. The orbital period was a feature that was provided, which is a very telling sign of what the distance would be: longer orbital period either means farther distance or faster orbit, and gravity keeps speeds fairly constant. To a large extent, we would feel comfortable using a machine learning model to predict the distance from a planet’s star.

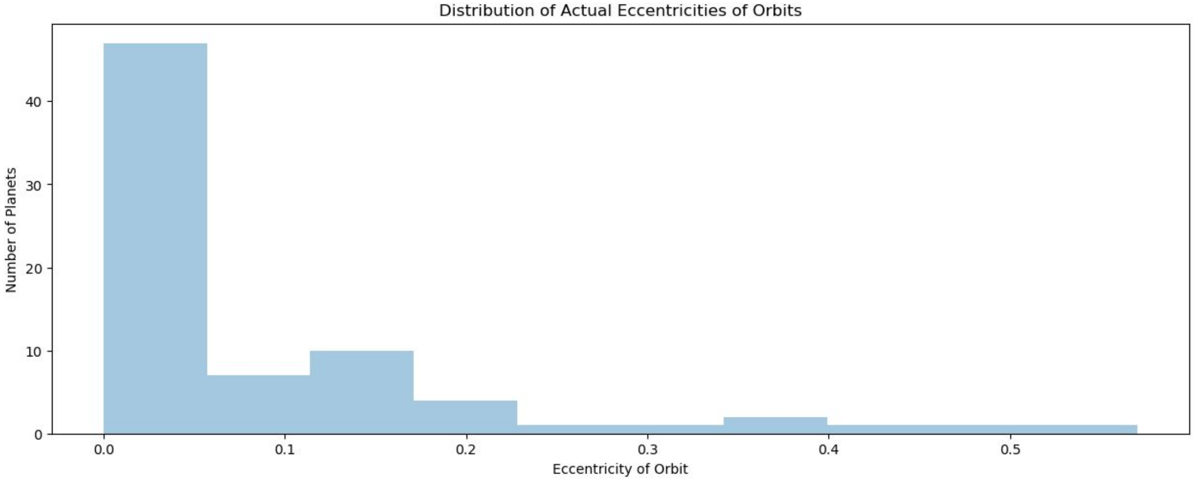




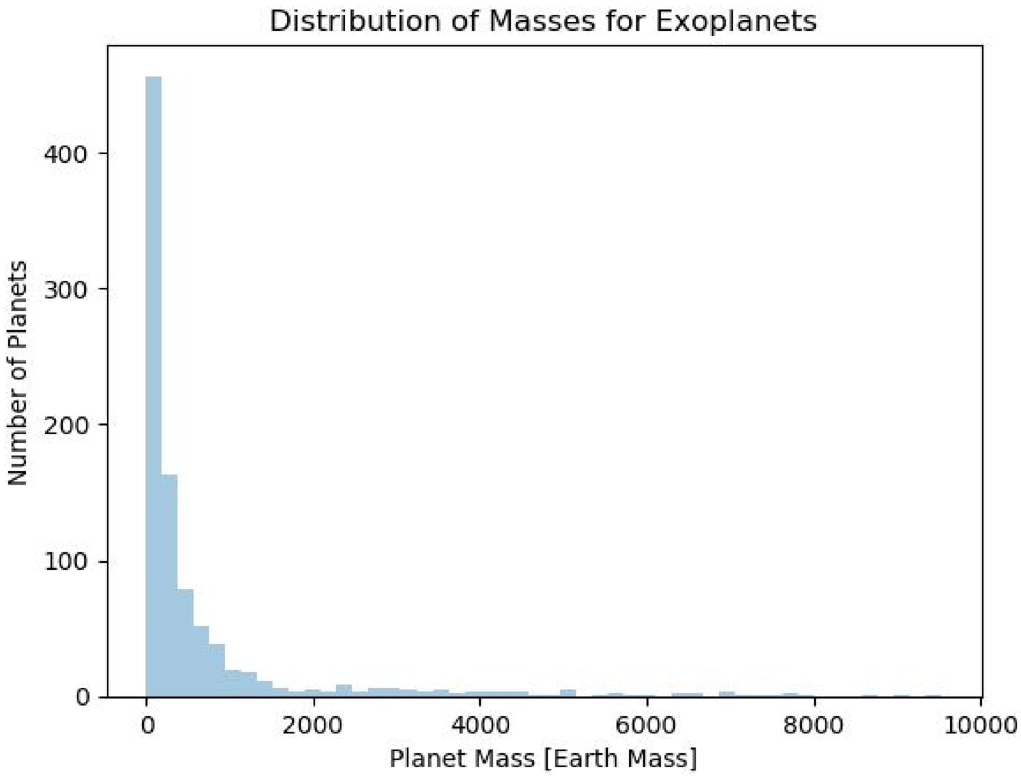
7. **How accurately can you predict the eccentricity of an exoplanet’s orbit? (ML)**

For this question, we were trying to see how accurate a machine learning model could be at predicting the planets’ orbital eccentricity given certain information. Here were the features we provided: number of planets in the solar system, planet’s orbital period, planet’s distance from its star, planet’s radius, planet’s mass, planet’s density, the star’s temperature, the star’s mass, and the star’s radius. Overall we would say that the machine learning model was not effective at predicting a planet’s eccentricity. Aside from the general ‘L’ shape on the left side of the plot, the visualization that the machine learning model produced does not match that of the actual data very well. The statistical values reflect this conclusion. The distance calculations had an average accuracy of 0.110192, and the r-squared value was -0.706243. 0.1 with a scale of 0.5 is enormous: on average the model was 20% off. Along with that, a negative r-squared value is very bad and is evidence of there being little to no correlation in the data. So, at least with our current features, we would not trust machine learning models to predict eccentricities of planets' orbits.

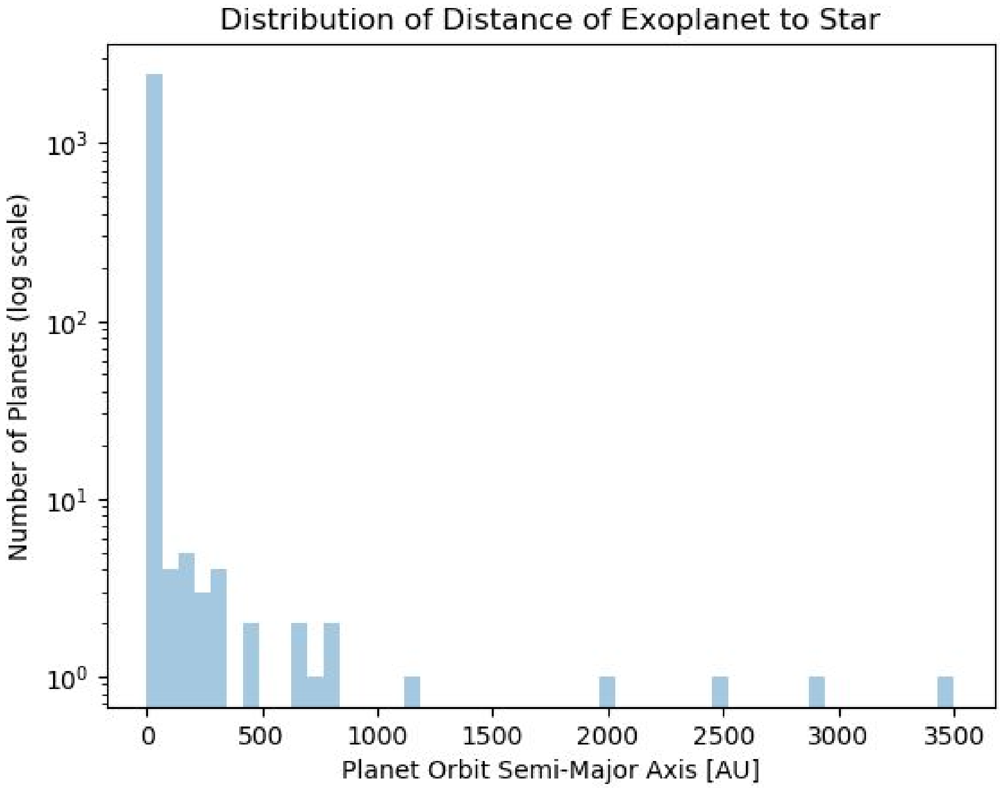




8. **What is the distribution of the mass of the exoplanets? (Graphing)**
For this question we used a distribution plot to show how many planets fall into a certain category of masses. We can see from the plot that a majority of the planets in our dataset have lower masses. As we increase the mass of the planet, fewer and fewer planets seem to fall into that mass category. This is expected because not all planets are the size of Jupiter or bigger. Looking at the raw data we can confirm this graph is correct. The raw data showed that a majority of masses fell between a couple 100 times the Earth's mass. Therefore seeing the data grouped by the start of the graph makes sense.



9. **What is the distribution of the distance from the star for the exoplanets? (Graphing)**
This graph shows the distribution of distance for exoplanets and how many planets fall into the specific distance category. We can see that a majority of planets have a smaller semi-major axis. The semi-major axis is the radius of a planet's orbit. There are some outliers if you look closely around the 2000 AU to 3500 AU range but we still see that almost all the planets have a smaller semi-major axis. Additionally, when looking at the raw data we can confirm this graph is correct. Most of the semi-major axes in the dataset were very small. Most being less than 1 AU but greater than 0 AU. Therefore, it makes sense to see most of the data grouped by 0 AU.



Challenge Goals

- **Machine Learning:**
    - We're going to use machine learning to answer questions, predict values, and predict the accuracy of our models. This is to see if machine learning is a viable option to predict missing values in our data, we will primarily focus on model accuracy.
- **Many Perspectives:**
    - We have a wide range of questions that we're answering about our topic and we believe this diverse set of questions meets the criteria for the "many perspectives" challenge goal. We can put them into a few categories:
        i. Basic information about exoplanets: This is information we find interesting and believe would be interesting to others. The motivation of our project is to inform people on exoplanets. This is a good way to provide basic exoplanet information that anyone can understand.
        ii. Machine Learning: We're trying to see if machine learning is a viable option to predict missing values in our dataset.
        iii. Search for life: We have a 2-step process for finding planets that are suitable for life. First is to determine which planets are in the

Goldilocks/Habitable zone. The second is to further filter planets on other factors such as planet density and eccentricity to determine if a planet can be deemed habitable for life.

Work Plan Evaluation

Tasks: Each task will be in its own separate python document.

1. **Pandas Coding and Report (Ani)**
    a. Estimated Time: 2 hours
    b. Questions: 1, 2
2. **Machine Learning and Report (Bradley)**
    a. Estimated Time: 1 hour
    b. One function per research question (5, 6, 7)
3. **Graphing and Report (Ani)**
    a. Estimated Time: 2 hours
    b. Questions: 8, 9
4. **Question 3-4 (Habitable zone + Potential for life):**
    a. **Bradley** does Goldilocks zone calculations and other calculations (planet density, eccentricity). Bradley will write the report on how the calculations are done.
        i. Estimated Time: 3 hours
    b. **Ani** does pandas and graphing once the calculations are done. Ani will write the report on how pandas and graphing was done.
        i. Estimated Time: 2 hours

Evaluation (Ani): For the pandas file my estimates were not accurate because it took me less time that I thought to complete the code for it. Since we used pandas and there were no loops, the time it took was cut in half. For the graphing file, the code took more time than I expected because I ran into some problems. Some of the graphs were not showing up as we expected them to because we were not using the correct seaborn plots. In the end Bradly and I worked together to figure out a way to get the graphs working and which visualizations were correct to use.. Last, the habitable zone code took me a little more time to finish. We had to rewrite some statements in our functions because it wouldn't work with a boolean Series.

Evaluation (Bradley): My evaluation of how long my parts would take were pretty underestimated. Doubling or tripling my predictions of 3 hours and 1 hour are more accurate. For the habitable planets file, it took longer than expected because I spent hours researching how to calculate planet temperature, doing dimensional analysis to figure out why my first equation was way off, then tweaking the working equation to return values that were closest to NASA's. I just eyeballed the accuracy rather than building a test to figure out the most effective number to multiply my result by. The machine learning file would have been an accurate time estimate had I not felt the mean squared error was insufficient at determining model accuracy for our project. So I spent the extra time adding mean absolute error, r-squared, and especially creating those dang plots which I ran into so many issues making. I also spent ~30 minutes working on finishing the work Ani did to create the habitable planet scatter plot because I had a specific look in mind which we felt I could best replicate.

Testing

For each file there was a function in our ‘main.py’ file dedicated to testing it.

Pandas File Testing:

For the pandas file, it was difficult to test because we had so much data, although using a smaller subset of the data it would be easier to do so. Therefore I manually did the calculations for 10 rows of the dataset and compared them to the values the code gave me. My manual calculations and the code matched, showing that the functions would work for the larger dataset. In the ‘Files’ folder is where you will find the file ‘Manual_Calculations_Pandas.docx’ and that file contains the manual calculations I did.

Graphing File Testing:

For our graphing file, all we did was plot points. We can trust that the distribution plots are correct because we can look at our raw data and see where there is a correlation between it and the data we graphed.

Machine Learning File Testing:

For our machine learning file, we ran 3 different machine learning models on our data to see how well machine learning can predict planet masses, distances from their star, and orbital eccentricity. After seeing the results, we didn’t feel that the mean absolute error (MAE) and r-squared value were enough to determine how accurate the model was, so we created a plot comparing the predicted values to the actual values to give us one more measuring stick for our model.

Habitable Planets File Testing:

For our habitable planets file, we created 2 functions to determine habitability for a planet, as well as 2 sub-functions with all the calculations behind the pandas work. The two main functions were tested by merely running them looking at the plot and comparing a few individual results to the dataset. The real testing was needed in the sub-functions doing the calculations. The function to determine the effective temperature of a planet was so complex that we created an additional function to print the results of the function and compare it to some effective planet temperatures we had access to. Once we were satisfied with the accuracy of the function, we left the printing function and our tests to be played with or examined by whoever looks at our code.

Collaboration

We both used git to collaborate and used our local machines to write our python code. We communicated on a regular basis on what we are doing and had access to what the other person is doing through git.

Other than git, we used some websites to figure out how to do the calculations for question 3 and 4. We also used some websites to do research on our results. All sources are linked in the report.