Shaping the FINAL FRONTIER

The future of orbital satellites



Wiley Jones B.S. Computer Science

[jonewile@oregonstate.edu](mailto:jonewile@oregonstate.edu)

CS 332

TAble of  
Contents

INTRODUCTION 1

Topic Information 1

Questions 3

DATA GATHERING 5

Direct Data Downloads 5

UCS Satellite DataBase 5

Space Exploration and Economic Growth: New Issues and Horizons 5

Rocket Emissions Impact Dataset 6

Global 3D Rocket Launch and Re-Entry Air Pollutant 7

Data Gathering Using an API 8

Kaggle: Mission Launches 8

DATA CLEANING 12

UCS Satellite Database 12

Space Exploration and Economic Growth: New Issues and Horizons 23

Rocket Emissions Impact Dataset 30

Global 3D Rocket Launch and Re-Entry Air Pollutant 36

UNSUPERVISED LEARNING WITH KMEANS 39

UCS Satellite Database 39

Space Exploration and Economic Growth: New Issues and Horizons 49

Global 3D Rocket Launch and Re-Entry Air Pollutant 61

SUPERVISED LEARNING WITH DECISION TREES 73

UCS Satellite Database 73

Space Exploration and Economic Growth: New Issues and Horizons 83

Global 3D Rocket Launch and Re-Entry Air Pollutant 91

CONCLUSIONS 99

# INTRODUCTION

## Topic Information

Whether it’s cutting-edge research into the depths of space or enabling the internet so you can look up the best burrito place on your cell phone, almost every facet of modern life is dependent on satellites and the technology they enable. The rise of orbital satellites has gone hand in hand with the advancements of the 20th and 21st centuries, and there is no sign of that trend slowing down. In fact, the space and satellite industries are only growing larger and more integrated into society with the rise of civilian companies in the industry. Therefore, a thorough understanding of the satellite market is needed to allow government and business interests to make the best decisions on continued launches of satellites into orbit.

As this growth continues, this report seeks to answer those fundamental questions about the present and future of satellites. There are currently over six thousand active satellites in orbit—all in various capacities, sizes, lifespans, and costs. I aim to illustrate the recent advances in the industry and illuminate where it is going in the future. These marvels of modern engineering are the backbone of the technological world and discovering the trends in the spread of these machines will reveal what we can expect from the world of tomorrow. As the planet continues to blast its destiny out of the atmosphere the potential for business and market growth is out of this world.

Nasa Launches GOES-U Satellite

As satellites continue to revolutionize industries and expand their presence in orbit, it is crucial to consider the environmental impact of their launches. Every rocket launch releases emissions into the atmosphere, including carbon dioxide, black carbon, and other pollutants that can affect both air quality and climate patterns. The growing number of satellite launches, especially with the rise of commercial space companies, raises concerns about the long-term effects of these emissions on the planet. Additionally, discarded rocket stages and defunct satellites contribute to the issue of space debris, which poses a risk to both existing satellites and future missions. Understanding these environmental impacts is essential for developing sustainable practices in the satellite industry, such as cleaner propulsion systems, reusable rockets, and responsible satellite deorbiting strategies. While the benefits of satellites are undeniable, ensuring that their growth does not come at the cost of environmental harm will be a key challenge in the years ahead.

This topic was chosen because many people are unaware of the impact that satellite technology has on their daily lives and the future of the world. As the number of launches only increases, I believe that developing an understanding of the industry is paramount to shaping a future where satellite technology is utilized to its greatest effect. It might be the difference in whether humanity’s meteoric economic and technological rise continues or fizzles out.

## Questions

What is the expected rate of growth in satellite launches? And when would the theoretical limit for low earth orbit be met?

* I would track the rate of increase in launches and project that into the future with the machine learning model. Then I would check that against the suspected limit of orbits possible.

How has the utility of satellites changed over time and what trends to expect in the future?

* Track the different uses of satellites over time and see what industries are expanding the fastest. Use the machine model to predict if the expansion is increasing or slowing for each industry.

What industry contractors are unserved and prime for expansion?

* See what industries are already heavily involved in the satellite space, and then map the companies or contractors in that space that have the least foothold. Could use some of the other answers to make suggestions on how to scale for them.

How will the costs of a satellite launch change in the future?

* Map the cost of satellites over time (might have to adjust for inflation) and forecast the future of prices. Would use the machine learning linear regression to make a ballpark prediction of the upcoming costs.

How has the weight of satellites changed and what are they projected to be in the future?

* Track the change of weights in satellites over time and forecast where it will go in the future. This would give insight into the future of the industry to see if more massive satellites are becoming viable with the costs.

How has the life expectancy of satellites changed and what will they be in the future?

* Track the life of active and retired satellites and use the machine learning model to find if there is a pattern in the longevity of the spacecraft.

Which countries will have the fastest-growing satellite networks? And who are the major players of tomorrow?

* Track the number of launches from various countries around the globe and see where the most growth is occurring. Use the machine learning model to predict expansion patterns for countries. Maybe cross-reference these trends with economic data to see how they correlate.

What launching stations will see the most growth?

* Track the number of launches from various sites and figure out how the numbers have changed. Uses this data to find what sites have the capacity for more launches based on the current rate of expansion.

What time of year is best for launches? And what areas in the world have the best conditions?

* Use the data to track the launch dates of various rockets and compare that to average weather data from the area. Cross-reference this with other areas with similar weather to see where new sites could be built.

Which orbits serve what industries better and what has room for growth?

* Document satellite uses in the different levels of orbit and see which industries are using what orbits. If an industry is using multiple orbits, predict if others in the industry should move to a different orbit with less competition for space.

# Data GatherinG

Direct Download

## UCS SatelLiTte DataBase

A screenshot of a computer

Description automatically generatedThe first dataset gathered was the UCS (Union of Concerned Scientists) Satellite Database. It is a comprehensive list of all the active satellites in orbit with data with technical specs of the spacecraft and information on the use, industry, company, and country of origin. It is a labeled CSV file dataset with both quantitative and qualitative data. To download this data set, google search “UCS satellite database”. The first result will be the official website for the UCS. On the page there are four download links. I choose the Database (Excel format). After downloading, it needs to be converted into a csv file.

View of download links on UCS website

**Quantitative Data**

Longitude of GEO, Perigee, Apogee, Eccentricity, Inclination in degrees, Period, Launch Mass, Dry Mass, Power, Launch Date, Expected Lifetime, COSPAR number, and NORAD Number

**Qualitative Data**

Name of Satellite, Current Official Name, Country of UN Registry, Owner/Operator, Users, Purpose, Detailed Purpose, Class of Orbit, Type of Orbit, Contractor, Country of Contractor, Launch Site, and Launch Vehicle

A table with numbers and a number on it

Description automatically generated  
Source: <https://www.ucsusa.org/resources/satellite-database>

## SPACE EXPLORATION AND ECONOMIC GROWTH: NEW ISSUES AND HORIZONS

This dataset is from an academic paper by Akhil Rao, Luisa Corrado, and Maureen L. Cropper for Middlebury College in Vermont. They wrote a paper similar to the aim of this one, on the economic development of the space industry with a heavy focus on the cost of sending objects into space. There are many datasets in the replication package, but the one to focus on is the “launch\_df\_base.csv” file. This file was the main one where the data was processed, so it is a collection of all the different datasets combined. It comprises a list of all launches into space with their date, name, rocket type, and much more. It is mixed, labelled dataset with qualitative and quantitative data.

**A screenshot of a phone

Description automatically generated**To download the package, first search “satellite launches price” in Google Dataset search. Then click the option “free” from the top bar. The first result will be the replication package for this dataset. The link takes you to the Middlebury College site. At the top is a “Download” button.

Link to site on google datasets

**Quantitative Data**

JCAT, Price Per Kilogram, Max Mass

**Qualitative Data**

Launch Date, Piece, Name, PLName, Launch Vehicle Type, Flight ID, Platform, Launch Site, Launch Pad, Ascent Site, Ascent Pad, Agency, and Launch Code

A screenshot of a table

Description automatically generated

Source: [*https://middlebury.figshare.com/articles/dataset/Calculating\_Launch\_Price\_Indices\_1957-2023/23992431?file=42076353*](https://middlebury.figshare.com/articles/dataset/Calculating_Launch_Price_Indices_1957-2023/23992431?file=42076353)

## Rocket Emissions Impact Dataset

This dataset is from an academic paper by Robert Ryan and Eloise Marais. It is a dataset that contains details of 2019 rocket launch emissions inventory included in GEOS-Chem simulations. It basely estimates the amount of emissions put into the atmosphere by each launch in 2019. It is a mix of quantitative and qualitative data that is labelled on whether the reentry was controlled or not.

**Quantitative Data**

Time (UTC), Day, Month, Year, Latitude, Longitude, Booster separation altitude, Booster Mass, First Stage Mass, Second Stage Mass, Total Mass, NOx from re-entry

**Qualitative Data**

Controlled, Name, Type, Part

A screenshot of a computer

Description automatically generated

Source: <https://rdr.ucl.ac.uk/articles/dataset/Rocket_emissions_impact_dataset/16974166>

## Global Rocket Launch and Re-Entry Air Pollitant and C02 emissions

This dataset is from an academic paper by Connor R. Barker, Eloise A. Marais, and Jonathan C. McDowell breaking down the emissions from launches of mega-constellation satellites from 202-2022. Mega-constellation satellites are launches where one rocket will set multiple satellites into orbit at once. It is labeled data on whether the launches are mega-constellational or not with a mix of qualitative and quantitative data.

**Quantitative Data**

Mega-constellational Flag, Rocket Name, COSPAR ID, Time (UTC), Date, DISCOSweb Rocket ID, Booster Fuel Type, Stage1 Fuel Type, Stage2 Fuel Type, Stage 3 Fuel Type, Stage 4 Fuel Type, Proxy Rocket

**Qualitative Data**

Longitude, Latitude, Booster Number, Booster PropMass, Booster StageMass, Stage1 Prop Mass, Stage 2 Stage Mass, Stage 3 Stage Mass, Stage 4 Stage Mass, Fairing Mass

A screenshot of a computer

Description automatically generated

Source: <https://www.nature.com/articles/s41597-024-03910-z#Sec7>

API Collected Data

## Kaggle: Mission launches

This dataset is hosted on Kaggle but compiled from the website spaceflightnow.com. It is a list of all launches into space, satellite or manned, from Sputnik to 2020. It has information on the organization that launched the spacecraft, the site, the date of launch, the name and rocket type, the cost of the launch, the rocket status, and mission success or failure. It is a CSV file mixed dataset with both quantitative and qualitative data. In addition, it is a labeled dataset because it files each mission into the status section—where missions are grouped between success, failure, or partial failure. This dataset was downloaded using Kaggle’s API (please see explanation after dataset general information).

**Quantitative Data Labels**

Launch Date and Mission Price

**Qualitative Data Labels**

A screenshot of a computer

Description automatically generatedOrganization, Location, Date, Detail, Rocket Status, and Mission Status

File link: <https://drive.google.com/file/d/15ChPXfLOOt56btpf6rJsH-i3UmPJ0RD2/view?usp=sharing>

\*\* I explain how to get from the API below but wanted to include a link as well.

**Connecting to the API**

First, log in or create a Kaggle account. Go to account settings and look for Create New API Token. A kaggle.json file will be downloaded that contains your username and API key. Open the file and save the username and key in a separate folder as well so you can access it.

The kaggle.json file needs to be put in a specific folder to be accessible by the Kaggle API. Therefore, the json file must be moved to:

* C:\Users\<YourUsername>\.kaggle\. for windows users
* ~/.kaggle/. for Mac/Linux.

Move the file into the .kaggle folder ether by copying the file path and pasting it into the computer directory or using the terminal to move it into the right folder. Also run the below command to make sure your permissions are correct.

* mv <YourFileLocation>/kaggle.json ~/.kaggle/
* chmod 600 ~/.kaggle/kaggle.json

Next, open python and import requests and KaggleApi. Save KaggleApi() to a variable (e.g. api) and authenticate with api.authenicate() method.

Save username, key, and endpoints as variables. Make a dictionary for the parameters, so they are already in key/value pairs. Enter parameters.

Use the requests get method with the endpoint, username, key, and params to get a list of datasets with your query. Save the response as a json. Print it out and it should look like the below.

A screen shot of a computer screen

Description automatically generated

Looking closely, there are two keys in the json file called “title” and “url”. Iterate through the json file by dataset and print just the titles and URLs. Should look like the below.

A screen shot of a computer

Description automatically generated

Take the first result and take the last part of the URL: sefercanapaydn/mission-launches. This is the dataset id. Save that to a variable. Use the Kaggle Api method “dataset list files” to see what the name of the csv file in the dataset. Should be mission\_launches.csv.



Use the Kaggle Api method “dataset download file” with the dataset id and name of the file. Finally, to check that the download was successful, use the open method to read the file into back into python and print the contents.

A screen shot of a computer screen

Description automatically generated

API URL: <https://www.kaggle.com/api/v1/datasets/list>

Example Query Url: <https://wileyalanjones:e1ba1ef1524d589752d2f17bcf8cb92e@www.kaggle.com/api/v1/datasets/list?search=space+missions>

\*\*Note this query will not work because the authentication doesn’t happen in the URL for Kaggle’s API. It is hidden from the URL, so can’t be accessed with authenticating from API.

A screenshot of a computer

Description automatically generated

# DATA CLEANING

## UCS SatelLiTte DataBase

This Dataset is a bit unwieldy to show because there are over 20 columns of data and over 60 columns total (though most are just empty). Here is a sample picture which does a good job of showing the kinds of information inside it.

A screenshot of a table

Description automatically generated

There are a mix of quantitative and qualitative variables, many repeated values per row, and a lot of missing values. So overall, there is a heavy amount of cleaning to be done for it. First on the cleaning list, is to get rid of the empty columns in the dataset. If we look at the list of column names from the dataset, there are many empty unnecessary columns at the end. Look at all the Unnamed ones in the picture below.

A screenshot of a computer program

Description automatically generated

A computer screen shot of white text

Description automatically generatedTherefore, removing these columns is the first cleaning step. After this step, we finally have a data frame that is just of the columns with data on them.

However, even now we can trim the columns further based the analysis we want to do. Looking at the columns left above, there are many that are useful in records keeping (which is what this dataset is mostly updated for) such as name, Org of UN Registry, COSPAR Number, and NORAD Number that we don’t need in our analysis. We will drop of those columns as well.

A screen shot of a computer

Description automatically generated

Now, it is time to look at the data types of the remaining columns. Printing out the data in python produces the below.

A screenshot of a computer program

Description automatically generated

Now a lot of these are correct, but there are a few ones that are off. The most obvious are the quantitative variables that are reading as objects. Those include Perigee, Apogee, Period, Launch Mass, Dry Mass, and Power. To get these in the right data type, we need to convert all these values to numeric types. However, Perigee, Apogee, Period, and Launch Mass all have values with commas in them, which means if we simply use the to\_numeric method on Pandas, those values will be set to NaN. So first we must convert all the values in those columns to have no commas, then perform the to\_numeric.

A screenshot of a computer program

Description automatically generated

For Dry Mass and Power, we need to check if those columns are worth having in the data. If you take a quick look at the csv file, both seem to be majority missing values.

A black background with white text

Description automatically generated

Doing some calculations, Dry Mass and Power only have a little over 10 percent of the rows filled in. Therefore, we are going to remove them. Also going to remove Detailed Purpose and Type of Orbit for similar reasons.

A computer screen shot of a program

Description automatically generated

The next datatype to fix is the Launch Date. We don’t need the exact date of the launch. All we need is the year of the launch to help track changes over time. So, for that we will convert the last two digits of each launch date to a year and then change the name of the column.

A screenshot of a computer program

Description automatically generated

Lastly, we want to change the Class of Orbit from an object into a category since there are only a few possibilities for each Satellite.

A computer screen shot of a program

Description automatically generated

After all of that, we are finally able to move on to missing values. We can use a loop to get all the missing values from each column to see which ones need the most attention.

A screenshot of a computer

Description automatically generated

From the graphic above, Expected Lifetime with over 2100 missing values is by the far the one that needs to the most addressing. All the others have negligible amounts of missing values compared to it. Deciding what to do with these missing values was tough. I thought about just deleting the values, but in the end, I didn’t want to give up all the other data that would be lost by deleting. But what value to fill the dataset in with.

A graph with a blue line

Description automatically generated

From the boxplot above, I think there is a heavy concentration of values near the mean and median with a few big outliers. Because of this, I chose to fill all the missing values with the mean value of 5.35. After that, I dropped all the other rows with missing values because the other columns had very little missing data. So, deleting them didn’t damage the integrity of the data at all.

Next step was to fix any incorrect values. This involves doing some initial visualization of the data to see where the mistakes are.

For example, on the Class of Orbit column, looking at this category plot shows that there was a mistake for one of the entries reading LEo instead of LEO.

A graph of blue lines

Description automatically generated

This is an easy fix by finding the value in question and changing it back to LEO. The graph below shows that fixed.

A graph of a number of blue lines

Description automatically generated with medium confidence

If we look at the Users data counts, we can see that there are two values at the bottom that should be part of the main Commercial and Government sections, but they have an extra space on them.

A screenshot of a computer

Description automatically generated

If we use the strip method on the values in this row, we see that this fixes the issue and the two are in the correct format now.

A screenshot of a computer program

Description automatically generated

Also, I wanted to make sure that all of orbits were correctly marked, so I created the boxplot below.

A graph of a diagram

Description automatically generated with medium confidence

There are two clear mistakes. One of the LEO satellites has a perigee that’s too far out. Perigee if you don’t know is the distance where the satellite is closest to its orbiting body (i.e. Earth). LEO means Low Earth Orbit, so having a Perigee over 10000 doesn’t make sense. Locating the row in the data frame, this entry was supposed to be a GEO satellite but was marked incorrectly. Also, there is a GEO orbit that has a Perigee that’s way too small. That data point can be drop from the data.

A graph of different colored objects

Description automatically generated with medium confidence

Also, if we look at the opposite chart of Apogee, which is the point that the satellite is furthest from the Earth, we see another outlier mistake.

A graph of a number of numbers and a number of numbers

Description automatically generated with medium confidence

This value can be dropped as well. It makes no sense that a GEO orbit which is none elliptical would have an apogee that far out.

A graph of a diagram

Description automatically generated with medium confidence

After all of that, the dataset is finally clean and ready for use in analysis. However, we still want to move the Class of Orbit column to the front of the data frame because that is now acting as our label for the dataset. Once that is done, read the new data frame to an CSV file for use later. Here is a picture of the result.

A screenshot of a computer

Description automatically generated

Now that we are finished with the main CSV file, I want to use this file to create a new data frame that has just quantitative and unlabeled data to use for unsupervised learning. For this new dataset, I decided to use Perigee, Apogee, Eccentricity, Inclination, Period, Launch Mass, and Expected Lifetime as my quantitative values. Here is a picture of the new dataset below.

A table with numbers and a number of people

Description automatically generated with medium confidence

## SPACE EXPLORATION AND ECONOMIC GROWTH: NEW ISSUES AND HORIZONS

This dataset is very different than the UCS database for this project. The UCS was a more general overview dataset for satellites with more technical and categorical data. However, for this dataset, we are focusing entirely on cost of sending satellites into orbit. Here is picture of the uncleaned data.

A screenshot of a computer

Description automatically generated

As you can see there is a lot of qualitative data about the different launches, but we are only going to focus on the cost, max payload weight, launch date, and vehicle type. First let’s look at the columns.

A black background with white text

Description automatically generated

There are a lot that we don’t want, so let’s drop all the columns that are not needed.

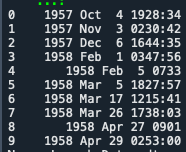


Now that that is complete, we can check the datatypes of the remaining variables.

A screenshot of a computer

Description automatically generated

Most of these variables look good, except for the Launch Date which we want to change to a date time. To do this, we need to change the format of the original column because it uses abbreviations of the months instead of numbers.



So, we need to change the format into year-month-day so pandas can change it to a time correctly. We can save the current column into a list and then run some string manipulation on them to put everything in the correct format.

A screenshot of a computer screen

Description automatically generated

After we are done with that, we run the pd.to\_datetime method on the column, and because it is in the correct format there are no errors.

A close up of a blue background

Description automatically generated

Next, we move on to addressing the missing values for the dataset. If we look at the missing data for all columns, we see that price\_per\_kg and max\_mass have similar missing values.

A black background with white text

Description automatically generated

However, the total number of rows in the dataset is over 19000, so removing the missing data for price and mass would only lose 15% of the total data which would still be over 16000 rows. Also, I should mention that the missing values for mass and price are almost exclusively the same. If one is missing on one row, it is extremely likely that it is missing on the other row as well. So, I believe it is fine to just drop all the rows with missing values from the dataset.

Now it is time to check for incorrect values in the data. For that we need to do some visualizations to see if anything is out of place.

A screenshot of a computer

Description automatically generated

From this printout of the counts of the LV Types, we can see that Space Shuttle missions are included in it. We are looking for only satellite launches because manned missions have different factors involved with them. We can remove these missions from the dataset. Also, we can remove all the apollo missions (Saturn V LV Type) as well for similar reasons.

A screenshot of a computer

Description automatically generated

Next, if we look at the price box plot, it looks like there are some crazy incorrect values.

A graph of a number of objects

Description automatically generated with medium confidence

However, I do think that these outliers are necessarily wrong. The large values at the extremes are for some of the first satellite missions ever in the 50s and 60s. Of course, back then, when the technology was new, the price to put things in orbit was astronomical, so it makes sense that the per kilogram prices are so high. Having said that, I think it’s ok to remove these from the price list as they are true outliers that don’t really reflect anything deep on the state the satellite market now.

A graph with numbers and a line

Description automatically generated

Even though there are still some large outliers on this new portion of the dataset, I believe they are much more manageable than the one before.

Now, before continuing, we need to add the label to our dataset. The issue is that we didn’t have a label on the initial dataset. However, there was another CSV file included in the files from the download that had the vehicle classification for all the launch vehicles. What we want to do is add that class label to our make dataset and create a new column for it.

A screenshot of a vehicle

Description automatically generated

So, to do this, we need to make a new data frame from the other csv file, drop all columns except the family and class, drop all the duplicates and Nulls from those columns, and then convert those two columns into a HashMap. From there we take out LV type column as a list, and then check each value against the HashMap. If the value is in the Family keys, we put the corresponding Class value into a new list. Once the new list is complete, we can add the list with all the classes to the front of the data frame as it’s label. Here is a look at the final product.

A table with numbers and a few different sizes

Description automatically generated with medium confidence

From here, we want to create a new data frame but normalize the quantitative values with using the min/max method. Here we only min/max the price and mass columns because those are the only quantitative ones we have. Here is the picture of the final output.

A table with numbers and letters

Description automatically generated

Finally, here is the new data frame with only the quantitative values for this dataset below. It is only the price\_per\_kg and max\_mass columns in the dataset.

A screenshot of a calculator

Description automatically generated

## Rocket Emissions Impact Dataset

I wanted to include this dataset because it talks about the emissions impact of Satellite’s on the atmosphere, so the focus here will be on the nature of the entry, mass of the object, and the amount of NOx released into the atmosphere. Here is a picture of the uncleaned data.

A screenshot of a computer

Description automatically generated

As you can see that this data is clearly record data in rows and columns with the label of Controlled as the first index. However, there are a few columns we need to get rid of.

A screen shot of a computer

Description automatically generated

Looking at the list above, we can see that we can get rid of Name, Part, and all the time columns because their information is not relevant for our analysis. In addition, Booster separation altitude, Booster mass, first stage mass, and second stage mass all don’t have enough values filled in to be relevant. After dropping those, we get the following columns remaining.



After dropping all those columns it’s time to check the data types for each of the remaining variables.

A black background with white text

Description automatically generated

All of these are mostly correct, but we can change the Controlled data type from object to category since it is the label for our data. Also, we should change the name of the variable from Controlled to LABEL.

A black background with white text

Description automatically generated

After that let’s check on the missing data for the data frame. However, there are none for this Data Frame.

A black background with white text

Description automatically generated

I think if there had been any missing values in the mass or NOx variables, I would have replaced them with the mean of the rest of the column. On the other hand, if there were missing values in the other variables, I would have considered just deleting them because I wouldn’t be sure what the best value to fill them in with would have been.

Next, we need to check if any of the values are incorrect or in the wrong format. Let’s start with the Label because that is the most important variable to have correct.

A blue rectangular bar graph

Description automatically generated

The good thing is that there are no stray values in the Label variable, but with the disparity in the absolute number of No’s and Yes’s, we might need to consider balancing the dataset—most likely with oversampling because the dataset is not large so under sampling would cause too much information to be lost.

Next, we can look at Longitude and Latitude together and see if any of the values are out of possible limit.

A diagram of blue and orange dots

Description automatically generated

If we look at this scatter plot of the data, there are no outliers or mistakes because all of totals are within the limit of Latitude and Longitude. Also, just something quick to note, all the positive re-entries seem to occur at a latitude between +20 and +40. Might be something to come back to in future analysis.

A graph with numbers and a bar chart

Description automatically generated with medium confidence

The above box plot of NOx shows that there are some outliers in the data, but I don’t think these are mistakes. This data is limited to only one year so it would make sense that there were a few rockets that produced more NOx then the rest. Therefore, they are going to stay in the data for now. If they seem to cause issues when we do analysis. I will come back and maybe change that decision.

A graph with numbers and a diagram

Description automatically generated with medium confidence

For the same reason, I am going to leave in the outliers for the mass section as well. There are not enough rows in the data to know for certain if these are mistakes.

With all of that completed, I am ready to read the now clean data frame to a csv file.

A screenshot of a graph

Description automatically generated

As a last step, we will remove all qualitative data from data frame to create a new one with just quantitative values. Those values will be Longitude, Latitude, Total Mass, and NOx from re-entry since those are the only quantitative variables in the data frame.

A table with numbers and a number of objects

Description automatically generated with medium confidence

## Global Rocket Launch and Re-Entry Air Pollitant and C02 emissions

This dataset is a mixture of three of dataset from the academic paper that I put together into a single file. First these were .nc files so I had to use xarray to import them as readable for pandas then convert them to data frames. Here are pictures of the three datasets.

A table with numbers and a black text

Description automatically generated

A screenshot of a data table

Description automatically generated

A screenshot of a computer

Description automatically generated

So this paper had used different files to cross reference each other in code to then get results. What I did was put the rocket attributes which had had the fuel type, mass, and booster numbers, into their own lists. And then created new columns with the launch dataset to fill in the information for all of them. After that I had a master data frame with all the info from the launch and rocket files. After that I totalled the mass of fuel used for each rocket by the five fuel types—Kerosene, Hypergolic, Solid, Hydrogen, and Methane masses into separate columns. After that, I used the emissions data file to calculate the total emissions per launch per greenhouse gas and added them to data frame.

A table of numbers and letters

Description automatically generated

Now we can get into cleaning the data frame. All the dtypes are right except for Booster Number which is reading as an object for some reason. I changed that to Integer.

A screenshot of a computer

Description automatically generated

There were 6 missing values in longitude and latitude that I dropped from data frame. I couldn’t find any mistakes. This was a very clean dataset already, and it is the smallest one that I have used so that would make more sense. The one problem I found was that Methane as fuel source was only used by one rocket in the whole dataset. I thought it best to take it out since one rogue data point might mess up any analysis that I am doing.

A graph of different types of objects

Description automatically generated with medium confidence

Lastly, I took the quantitative columns and made a new data frame with only those values.

A table with numbers and text

Description automatically generated

# Unsupervised Learning with KMeans Clustering

## UCS SatelLiite Database

A screen shot of a computer

Description automatically generated

Here is a picture of our cleaned CSV file from before. As you can see, there are many columns we want to drop to get ready for KMeans. We need to remove all the data that is not quantitative.

However, the first thing we need to do is save the data's label—Class of Orbit—to a list so we can reference it after the KMeans have been performed. After that, let’s delete all the columns not in the three we want to keep.

A screen shot of a computer screen

Description automatically generated

Now the data is ready to be used in KMeans, but first let’s get a visualization of the data we have at hand. The three variables, I am choosing to focus on are Period, Launch Mass, and Expected Lifetime.

Let’s put those on a plot and see the results.

A graph with numbers and dots

Description automatically generated with medium confidence

There are a lot of insights from this table. The first characteristic that jumps out is that most satellites are huddled around the low mass and low period end of the spectrum. This makes sense as most of the satellites launched today are in low earth orbit. LEO is the most functional for Earth-facing uses of satellites because it is closer to the surface of the planet meaning it’s easier to interact with instruments on the ground. What is unexpected from this visualization is that there doesn’t seem to be a big correlation between the Mass of a satellite at launch and its expected lifespan. There is a widespread of lifespans over all the masses. This is interesting because I would have believed that more massive satellites would be designed to last longer in space since the cost of launching increases with mass. This shows that there are more direct factors in describing why some satellites last longer than others than simply the mass of the launch.

Another interesting insight from this data is clearer if we invert the x-axis to have mass and period start at the same origin point. A graph of data visualization

Description automatically generated with medium confidence

This chart reveals that there is a correlation between the mass and period of a satellite—as the period increases so does the mass. I believe this also makes sense because satellites with longer periods—the total time it takes to orbit the earth—by design must be sent further out into space. The further out into space they are the longer the orbit will be. This is something that we all intuitively could understand but seeing it on a graph just drives home the point.

Now, let’s apply KMeans to the data prepared dataset. We will do clusters of 2, 3, and 4.

Two Clusters

I set up the KMeans object with 2 clusters and then ran the prepared data through the fit method. After that I saved the labels and the cluster centers into variables. Here are the print outs below.

Labels:

A screen shot of a computer code

Description automatically generated

Cluster Centers:

A number on a black background

Description automatically generatedA number on a black background

Description automatically generated

Here are the cluster centers plotting on the same chart as before. I’ve turned down the alpha way down on the dataset values so both centers are visible.

A graph of a graph with numbers and a diagram

Description automatically generated with medium confidence

I think these clusters are well positioned in the middle of the data. One of the stark divides that is clear from this new chart is that the two clusters are very far apart when it comes to the lifespan of a satellite, compared to the other variables. That might reveal that there is something about the differences in longevity that separates the different satellite groups.

Now let’s predict a couple new satellites and see where it falls on the labels. Our new satellites have the following specs.

Perigee: 10000   
Apogee: 10500  
Eccentricity: .1  
Inclination: 35  
Period: 5000  
Mass: 1000  
Lifespan: 6

Perigee: 100   
Apogee: 150  
Eccentricity: .6  
Inclination: 57  
Period: 90  
Mass: 200  
Lifespan: 2

Now we can use the KMeans predict method to see where each of these new datapoints would go.



Interesting enough they both go under the first cluster.

Three Clusters

Now let’s move on to 3 clusters for the dataset. I repeat my steps before, only this time, using 3 clusters. Here are the results from that.

Labels:

A screen shot of a computer screen

Description automatically generated

Clusters:

A number on a black background

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a computer code

Description automatically generated

From just looking at the printout of the labels, the third cluster is not nearly as represented as the first two. Also, the first two clusters stayed relatively the same. If we plot the results again.

A graph of a graph with numbers and dots

Description automatically generated with medium confidence

The third cluster is very randomly spaced then the other two. I think this shows that for the three variables chosen in the visualization, the two-cluster version is more aligned with the data. The third cluster must be more influenced by the other variables not on the chart.



Running our new satellite data through the three cluster predict method, gives the same results as last time.

Four Clusters

Repeating the steps again with 4 clusters, we get the following results.

Labels:

A screen shot of a computer code

Description automatically generated

Clusters:

A number on a black background

Description automatically generatedA computer screen with numbers

Description automatically generatedA number on a black background

Description automatically generatedA number on a black background

Description automatically generated

With four clusters, we see that the third and fourth clusters are even less represented in the data labels then before. Let’s look at the new plot.

A graph of a graph with numbers and a grid

Description automatically generated with medium confidence

From the plot four clusters are a randomly spread, and don’t give a lot of good insights on the data.



Doing predictions on our two new satellites also shows that they both would go in the first cluster.

Conclusions

From doing all the clustering of different k means steps, I think the one that fit the data the best was the two clusters. They have a more defined and realistic separation of the dataset. The larger the number of clusters the more random and unnatural the centroids were. I believe this is because most satellites are all in a similar with very similar values for all of the technical quantitative specs. This makes sense because LEO is by far the most popular of orbits for every industry. I believe what the two clusters were revealing was that many satellites fall under a LEO mean which was the first cluster. The other satellites which vary from the standard LEO orbit values can all be put into a second bucket of satellites. This second bucket is essentially saying, these are the satellites that are not part of the standard LEO orbit system.

For my project, I think this reveals that because of the number of LEO objects, doing any analysis should really be split into studying those standard LEOs, with the rest that don’t fit that mold being put in a special case area. For the next step, I think taking all the LEO satellites only and doing an analysis of just the differences between those objects would reveal more subtle differences. Right now, analyzing all together is not as helpful because the variance between the majority of LEO satellites and non-LEO satellites skews the data to just separating them out.

## SPACE EXPLORATION AND ECONOMIC GROWTH: NEW ISSUES AND HORIZONS

A screen shot of a computer

Description automatically generated

Here is a picture of the clean dataset that we are going to use for KMeans. First step is to remove all the non-number columns. We can do this with the select dtypes method to drop all non-numerical columns. Then we can drop the unwanted “Unnamed: 0” that is a residual from the CSV indexes.

A screen shot of a number

Description automatically generated

Now we only have two columns of data for this dataset, but we can create a three plot by multiplying the two numbers together to form a three column that would be called max cost—the total cost of the rocket if the maximum mass is used at launch.

A graph of a satellite cost chart

Description automatically generated From this chart, you can see that despite a few outliers, the variance of total cost compared to the Price Per Kilogram is not as pronounced a trend as expected. I would have thought that the larger cost per kilo would mean that the total cost would increase heavily. However, from this chart, the actual change is very little. I believe this is because as the technology has advanced, the price per kilo has dropped with the total mass of rockets going up. Therefore, the total cost of the launch remains more stable. However, when looking at Mass compared to total cost, there is more of a relationship there. It’s easier to see on a 2D plot.

A graph with blue dots

Description automatically generated

From this plot, we can see that in general, as the mass increases the total cost does as well. However, this increase is not very steep which illustrates my previous point that improvements in rocket technology have actual made the total cost stay relatively flat while the amount of mass being launched has increased.

Two Clusters

Labels:

A screen shot of a computer code

Description automatically generated

Cluster Centers:



These numbers correspond to the following variables:

1. Price per kilogram  
2. Max Mass  
3. Max Cost

I think from an initial look at the data, price per kilogram and total cost are the variables that vary the most between the two clusterings. This might be a sign that the biggest factor differentiating rockets might be their cost rather than mass. Let’s add them to the 3D plot we made. A graph of a satellite cost chart

Description automatically generated

The plot also shows that the two points are relatively close to one another. I think this shows that because of the rise of satellite launches in recent years, the rise of satellite launches means that the majority of the data is huddled around the modern numbers for mass and price per kilo—with the outliers being rockets from years ago with worse technology and fewer launches.

Let’s make some predictions using our new data. We will add two rockets with the following characteristics.

Max Mass: 3,000  
Price per Kilo: 8,000  
Max Cost: 24,000,000

Max Mass: 20,000  
Price per Kilo: 3,000  
Max Cost: 60,000,000

Let’s see what the predictions are for our current clusters. We make a list of list with the information above and then run it through the KMeans predict method to see which cluster KMeans would put it in.



Both got put into the second clusters with the lower of the masses and price per kilo gram.

Three Clusters

Moving the clusters to three, let’s see what the results give us.

Labels:

A screenshot of a computer screen

Description automatically generated

Clusters



Let’s look at these on the plot to see what it is trying to tell us.

A graph of a satellite cost chart

Description automatically generated

As you can see, the first two clusters stayed relatively the same, but the third one is much further out. If we look at what is around the new cluster. There are a few points with a lot of data around it. I think this new cluster is sitting as an outlier center. The first two clusters are more in the middle of the data where most of the data points are. However, the third one is not near the bulk of the data.

Let’s use the same launches we had last time to see if we get any different predictions. Running the predict method on our new launches we get that both are now in the first cluster.



This is interesting because it didn’t seem like the first two clusters moved as much in this iteration.

Four Clusters

Labels:

A screen shot of numbers

Description automatically generated

Clusters:





There’s a lot going on here so let’s plot the data. Also let’s see if we can divide up the inputs from the dataset into the different clusters to get a better visualization of them.

A graph of a satellite cost chart

Description automatically generated

What I think is great about this chart is that it clearly defines the pattern that the KMeans algorithm is revealing. The lumping together is happening more on the Z axis (Total Cost). It’s a little tough to see with this angle of the graph, but if we invert some of the axes, it becomes much more pronounced.

A graph of a satellite cost chart

Description automatically generated

This view gives a better perspective that as the total cost of the launch increases the more likely it is to move into a new cluster category.

Let’s now see which cluster our new launches would fall under in this version with four clusters. Here are the specs again to refresh your memory.

Max Mass: 3,000  
Price per Kilo: 8,000  
Max Cost: 24,000,000

Max Mass: 20,000  
Price per Kilo: 3,000  
Max Cost: 60,000,000



Not surprisingly the new entry we created with a higher total cost was put into one of the larger KMeans buckets.

Conclusions

A graph of a satellite cost chart

Description automatically generatedAfter concluding all the k values for the KMeans algorithm for this dataset, I think that one of the main takeaways is that the clusters were centering themselves much more around the price per kilo and cost variables than the mass variable. If we look at the way the clusters developed as more were added, there is somewhat of a pattern in the data. The first cluster is always a lower price-per-kilogram and lower total cost combination that progressively rises as we move up the clusters. The max mass, however, does not have this clear movement, and it can swing back and forth from the clusters being lower or higher than the previous one. This shows that the more defining variable in the data is probably the cost because it separates the data points more distinctly than the mass or even the price per kilogram.

If we look back at the last chart with four clusters. Look how the different colors are spread out on both the Mass and Price axes, especially the purple, which corresponds to the first cluster. It’s always a V shape where it has extremes sticking out in both high price per gram, low mass, and higher mass, but lower price per kilogram. In a regular 2d plot these would be opposite sides on the chart. Yet here they are lumped together. That reveals that what the total cost of the rocket becomes is more important to what cluster it reaches than these other two variables.

I think this is an important insight because it shows focusing solely on the total cost of a rocket launch might be the best metric to use. There are all these other variables and classifications that are valuable for their own use, but if we want to get base insights, cost might be the one that reveals the most. I look forward to diving into this more in the remainder of this report.

## Global Rocket Launch and Re-Entry Air Pollitant and C02 emissions

A black background with white numbers

Description automatically generated

Here is the initial look at the quantitative dataset that was created during the cleaning stage. It already has only quantitative data, so we are good to run it through KMeans already. This dataset has a ton of quantitative data, so here are all the column names.

A screen shot of a computer program

Description automatically generated

Let’s remove Longitude and Latitude. They are quantitative data, but they don’t factor into the analysis we want to provide. Before putting it into KMeans though, let’s visualize the data in 3D. Since this data is focusing on the emissions of rocket launches, I am focusing on the chemicals left behind after a launch, specifically CO2, CO, and BC (Soot).  
A graph of a satellite emissor

Description automatically generated

If we look at this chart, there are a few things that jump out. One is that there is a clear linear relationship between CO and CO2 emissions. That is not all that shocking as the two compounds are similar, only being one oxygen atom apart. If we look at a 2D linear regression plot of CO and CO2, we see the relationships clear.

A graph of a graph with blue dots

Description automatically generated with medium confidence

If fact, the first graph shows that there’s a linear relationship between all the three of these variables. Which is not too surprising since the larger the rocket the more emissions will come out. However, if we plotted this with different emissions, such as Nitrogen Oxide the relationship is not as clear.

A graph of a number of dots

Description automatically generated

There is a direct correlation between CO, CO2, and Black Carbon which is something to keep an eye on as our analysis continues. It will be interesting to see if the KMeans clustering keeps all this relationship when the clusters are added.

Two Clusters

Labels:

A screen shot of a computer screen

Description automatically generated

Clusters:

A screen shot of a computer

Description automatically generated

A computer screen shot of a person's head

Description automatically generated

There is a lot to unpack here, but if we focus on our three variables form before, the clusters have separated them into a low and high group. Although, it is strange that the group with the higher emissions actually has a lower Booster Number. That is something we should dive deeper into later.

A graph of a satellite emissor

Description automatically generated

Adding the clusters to the chart shows that the two clusters are keeping the same linear relationship that I pointed out in the initial visualizations. Now let’s create two new emissions profiles to add to make some predictions.

Booster No: 0  
Booster PropMass: 0  
Booster StageMass: 0  
Stage1 PropMass: 90,000  
Stage1 StageMass: 6,500  
Stage2 PropMass: 20,000  
Stage2 StageMass: 2,400  
Stage3 PropMass: 0  
Stage3 StageMass: 0  
Stage4 PropMass: 0  
Stage4 StageMass: 0  
Fairing Mass: 1000  
H20: 90,000,000  
H2: 1,000,000  
CO: 15,000,000  
CO2: 30,000,000  
BC: 4,000,000  
NOx: 0  
AI203: 0  
CLY: 0

Booster No: 4  
Booster PropMass: 150,000  
Booster StageMass: 14,000  
Stage1 PropMass: 400,000  
Stage1 StageMass: 25,000  
Stage2 PropMass: 100,000  
Stage2 StageMass: 4,000  
Stage3 PropMass: 5,000  
Stage3 StageMass: 900  
Stage4 PropMass: 0  
Stage4 StageMass: 0  
Fairing Mass: 1800  
H20: 150,000,000  
H2: 5,000,000  
CO: 250,000,000  
CO2: 300,000,000  
BC: 12,000,000  
NOx: 700,000  
AI203: 0  
CLY: 0

After we run both through the KMeans predict method, we get that the first one is in the first cluster and the second one is in the second cluster.



Three Clusters

Labels:

A screenshot of a computer screen

Description automatically generated

Clusters:

A screenshot of a computer

Description automatically generatedA computer screen shot of a person's head

Description automatically generatedA screen shot of a computer

Description automatically generated

Again, there is a lot going on with this dataset, so these clusters are hard to wrap our minds around. Let’s look at the chart with these cluster centers added.

A graph of a satellite emissor

Description automatically generated

Despite the chaos of the pure numbers, the graph does a good job of showing that the clusters are following the data well. It adheres to the linear relationship still. It feels like the clusters are showing where the data clumps up on the linear pattern. See how each X is in the middle of a distinct pile of data. This reveals there might be groups of rockets with emissions that all fall into similar buckets. If we look at the fuel makeup of their boosters and stages, we might be able to see which fuel is the best for emissions using this kind of analysis.

Let’s do our predictions for the two rocket emissions we set up before. These are the same two rockets now just being predicted with three clusters.



This time the predictions stay in the same place, so the first rocket goes in the first clusters and the second in the second.

Four Clusters

Labels:A number pattern with white numbers

Description automatically generated with medium confidence

Clusters:

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Let’s go straight to the graph to see if the linear relationship was holding true for four clusters.

A graph of a satellite emissor

Description automatically generated

Very interestingly, the new cluster point is not right on the line as the other three had been before. However, this one is in the secondary line that is like the main one but doesn’t have the same ratio of the three elements.

A graph of a graph with blue dots

Description automatically generated with medium confidence

If we look back at the linear regression plot that we made before, we can see this secondary line below the main plots very clearly. Therefore, this fourth cluster is considering this secondary uncurrent in the clusters.

Now, let’s do the predictions again for the two rockets we set up. 

This time the first rocket is put in the first second cluster and the second in the first.

Conclusions

After doing all the calculates of the KMeans, I think that the analysis reveals that the emissions data has a great effect in dividing the data. There were clear pockets of data that the clusters were centering around. Let’s look at the same graph from the 4 clusters but this time with coding for which datapoints are in which cluster.

A graph of a graph with different colored objects

Description automatically generated with medium confidence

This clearly shows that there are four distinct areas in the emissions data for CO, CO2, and BC. Look how defined the groupings are and the way that the clusters are right in the middle of these groups. Though surprisingly, the red grouping has two sections. A main one to the left of the center relationship line, but also a grouping near the origin. I think we can conclude from this data that the emissions spectrum of a rocket is dependent heavily on its fuel type and fuel mass. The reason for all these distinct groups is different rockets have different fuel types and levels of fuel use. These clusters are just showing which types fit together over all different types of rockets. Although, that’s not a deep insight, what is deeper is the way we can use this data to pinpoint which rocket fuels mixtures have the worse emissions outcomes and make changes based on it.

# SUPERVISEd Learning with Decision Trees

## UCS SatelLiite Database

A screen shot of a computer program

Description automatically generated

Here is a picture of our cleaned CSV file, read into Pandas as a data frame. As you can see it is not ready for Decisions Tree analysis because there is still qualitative data on it. We can see the labels in this picture which is the class of orbit. The first thing we must do is save the label to reattach after dropping the other columns.

A black background with white text

Description automatically generated

After that, let’s drop all the non-numeric columns to prepare for the decision tree. We will add the label back on after.

A screen shot of a computer

Description automatically generated

Now we have the desired columns for our decision trees analysis. However, we are not done with the data cleaning yet. If we look at the count for the labels of the data, we will see that it is very imbalanced.

A graph with different colored bars

Description automatically generated

To counteract this imbalance, we are going to have to do a mix of under-sampling and over-sampling to get a balanced dataset. We will under-sample the majority class, LEO, until it is balanced with GEO. To do this, we have to make a new data frame with only those two label options.

A screen shot of a computer

Description automatically generated

After that, we can use the Imblearn Random Under Sampler to get LEO and GEO balanced. Once that is done, we make another new data frame that is only MEO and Elliptical values and merge this new data frame with the resampled one of the other values.

A bar graph with numbers and a number

Description automatically generated with medium confidence

Now we are getting closer to balanced data, but not all the way there. Now we want to oversample the two minority labels to be equal to LEO and GEO. To do this, we will now use Imblearn Random Over Sampler to fill in the gaps. Here are the results are running the procedure.

A bar chart with different colors

Description automatically generated

Perfect! Now we are ready to split the data into the training and testing sets. We will use the Train Test Split method on our new balanced data frame to achieve this.

A screenshot of a computer

Description automatically generated

We now save the labels for the training and testing data in their own variables, drop them from the datasets, and now set to do our decision tree analysis. However, let’s take a deeper look at our training data first.

Here is a correlation matrix of the training set variable that can give some insight into how the decision tree might play out.

A graph of different colored squares

Description automatically generated with medium confidence

A correlation matrix shows how related two variables are to each other. A score of 1 means that they are perfectly correlated—if one is high the other will be high—and a score of negative one means that they are perfectly negatively correlated—if one is high the other is low. These relationships between the variables can show us how the tree might split the data when we run it through the decision tree. If we look at the above matrix, one of the points that stands out is apogee and period. This relationship makes sense as apogee is the highest point of a satellite’s orbit while period is it’s time taken to orbit the earth. Therefore, a high apogee means a longer orbit. This could reveal that the decision tree will most likely use these two variables early in the tree to divide the data between the four label points. Because of their close correlation, they could be instrumental in finding pure nodes since longer, higher orbits are important in classifying satellites. On the other end of the spectrum, inclination seems to be negatively correlated with every other variable. Inclination measures the angle between a satellite’s orbit and the equator of the Earth. This information is important for the researchers and businesses using the satellite but based on this matrix it might not be a metric that is all that useful in classifying orbits.

A graph with blue dots

Description automatically generated

Let’s dive deeper into the Period/Apogee relationship. If we look at the chart above, the linear relationship between Apogee and Period is abundantly clear. If we were to do KMeans on only this data, there would be definite clusters around the four times of orbits. The data fits too neatly for that not to be the case. Interestingly, now let’s look at Period to Perigee—the closest distance a satellite gets in its orbit.

A graph showing the difference between a period and a period

Description automatically generated

The relationship is much less clear. You can see the regression line has a very wide variance as it gets further and further. How can these two orbit length variables give us such different results? Well, the thing to remember is that most orbits are essentially circular, meaning the Apogee and Perigee have a small difference. However, there is a class of orbits that are ellipses. These orbits have an Apogee that is extremely far from Earth but a Perigee that can get as close as LEO. This class is one that is making Apogee have a steady linear relationship but throwing Perigee all out of whack. I think the decision tree will take this factor into account when figuring out the elliptical orbits.

Another correlated set of variables is Launch Mass and Expected Lifetime. Let’s take a closer look at relationships.

A graph of a diagram

Description automatically generated with medium confidence

From the joint graph above of the two variables, a couple of interesting details are illuminated. First, the vast majority of the data hovers around the low mass, low lifetime zone. This fact may not be surprising given that most satellites are LEO and generally lighter, but remember this data is from our balanced data frame. Theoretically, it shouldn’t have the bias that the non-balanced data has towards LEO. This reveals that lower lifetime and lower weight are still related even when we take the factor of the LEO satellites out of the equation. I think that is fascinating. Another insight is that there is a trend for satellites with longer lifespans to have a larger launch mass. It’s not incredibly clear in the graph but if you look at the way the clusters develop as the lifespan increases the launch mass does go up. This relationship is subtle but might be critical in the categorization from the decision trees model. Satellites with larger masses are usually sent further into space and therefore have longer orbits than the basic LEO. The decision tree algorithm will pick up on this fact in the training and make it a big part of the tree.

Decision Trees

Now it is time to run our data through the decision tree algorithm. We have our training and test data with the corresponding labels. We set up for Decision Tree object and set the max depth to three to make sure the tree isn’t too big. We get the below result.

A diagram of a data flow

Description automatically generated with medium confidence

This is the layout of the decision tree that was created with the testing data. Before we discuss, let’s test the accuracy with the testing data. Here is the confusion matrix with the results from the testing data.

A graph with blue squares and white text

Description automatically generated

This confusion matrix shows the decision tree model was incredibly accurate in predicting the correct classification of the orbit. Out of all the testing data, only three wrong answers were made. That means the model has an accuracy of over 99%. With this mode, we can classify the satellite orbits with extreme confidence. If you look at the decision tree above, you can see that the model starts by separating the satellites by their Perigee—closest point. The ones with the further out Perigees can only be GEO or Elliptical and in the next node of the tree that distinction is clarified by the period being under 2000 minutes. Anything longer is Elliptical and shorter is GEO. On the other side, the model uses Eccentricity to break off the remaining Ellipticals from the rest. Finally using Apogee—the furthest point—to separate the LEO and MEOs. This tree is a great representation of the power of supervised learning. With only five splits of the data, the model was able to be almost entirely accurate. It probably would have been even better if I didn’t limit the depth. Overall, we can feed any combination of the data points to this model and expect to get almost an entirely accurate classification of the corresponding orbit.

## SPACE EXPLORATION AND ECONOMIC GROWTH: NEW ISSUES AND HORIZONS

A screenshot of a computer screen

Description automatically generated

Here is the picture of our cleaned dataset from earlier. With only five variables, the cleaning for this data is relatively simple to prepare for the Decision Tree. First, we need to create our Max Cost variable from before by multiplying the Price Per KG by the Max Mass. After that, we save the label which is the Class variable, and drop the rest. Adding the label back into the data frame, we get the following result.

A screenshot of a computer screen

Description automatically generated

Now we need to investigate balancing the data. Let’s look at the totals for the Class variables.

A graph of a bar chart

Description automatically generated with medium confidence

From the chart above, it is easy to see that this dataset is imbalanced. The Medium classification dwarfs the other two in terms of numbers. However, small and heavy still have a decent number of total entries to process for the data. Because of that, we are just going to Under sample Small and Medium so that is matches the total of Heavy.

A graph of a number of labels

Description automatically generated

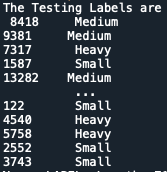
Now we have our dataset ready, we can split the entries into training and testing sets using the train test split method on pandas.

# A screenshot of a computer Description automatically generated

# A screenshot of a computer code Description automatically generated

Now we remove the labels from the split data and save them to use in our Decision Tree later.

A screen shot of a computer

Description automatically generated

Now that we have our training data let’s dig into it more. Let’s take a closer look into each of our three variables.

A graph of a number of values

Description automatically generated

Since we are looking to have the decision tree classify the Class of the Rocket Launch Vehicle, the mass of the payload would be a big factor in that. From this chart though, it appears that the training data has a big bias for values in the 0 to 25000 kilograms range. This screams to me that the training data must not be very balanced and is favoring small and medium over heavy. However, looking at the value counts of the training labels—which reveal how much of each label is in the training data—there is a very good mix between the variables.

A pie chart with numbers and text

Description automatically generatedHeavy is the most represented in the training data which is surprising after seeing our original chart. I think that this disconnect between the mass payload mass and classification will affect the decision tree model. The change in the technology is such that the early rockets with tiny payloads compared to today will throw off the decision tree algorithm because it’s a break from the logic pattern that less payload must mean the classification for the rocket is smaller. This is something I am going to be keeping an eye on as we proceed with the decision trees. Next, let’s look at our Price Per KG and Max Cost variables together.

A graph of a cost comparison

Description automatically generated with medium confidence

The thing that stands out most from this data is how random it feels. There are extreme outliers on both axes, but they lack a direct linear relationship. In fact, the outliers almost appear to have an inverse relationship, where the higher the price per kilogram, the max cost is stagnant or subtly decreases in some areas. Again, I believe this is from the fact that the data set has historical data on all launches dating back to the birth of spaceflight. Advances in technology have so drastically shifted the economics of space that looking at the eras all together produces these disjointed results. I can foresee the model again struggling here to find a reliable trend in the variables to get the correct classification. Overall, I’m worried that the nature of the dataset will make decision trees difficult, but only one way to find out.

Decision Trees

After we run the training data and labels through the Decision Tree Classifier algorithm we get the following tree.

A diagram of a data flow

Description automatically generated with medium confidence

Now we can test the model against our testing data to find its accuracy. After we run the testing data through the Decision Tree predict method, we get the following results.

A graph showing the size of a rocket economy

Description automatically generated

From looking at the chart, it was clear that my doubts were unfounded. Once again, the decision tree had an almost perfect accuracy in predicting the testing data correctly. Unsurprisingly, the main factor that it kept using to split the tree was the Mass of the payload. Even though I was worried about the seeming imbalance in the data, the mass still was the most pertinent factor in deciding the classification of the tree. Price Per Kilogram and Max Cost were only used once each in the five splits in the tree. What I find interesting is that the one place where the data had the most mistakes was categorizing Medium class rockets as small. This is not something that I would have suspected, but as I thought about it, I reconsidered. Since Heavy is the top end, it makes sense that they would be so much bigger than the rest that they are easily differentiated while the buffer between medium and small is more fluid. I would love to see where this model goes in the next decade as space flight becomes more abundant and cheaper. It could change the details so much that the model would have to be retrained on the new data. For my project, this model can be used to classify any new rockets that are being tested. If the engineers are hoping to find a way to get more payload into a small rocket, this model could be used to show them if their new design is registering as a medium. That would be a good sign for them as it would imply that the new small rocket has the same qualities as a medium in a lesser package. Overall, I’m very pleased with the results of the Decision Tree for this dataset that I was skeptical of.

## Global Rocket Launch and Re-Entry Air Pollitant and C02 emissions

**A screenshot of a computer code

Description automatically generated**Here is an image of the initial data frame. As you can see it still has many qualitative variables that need to be removed from the data. First, we save the label and then remove all the non-numeric data type columns from the data. In addition, we will remove Longitude and Latitude as well since they don’t have any bearing on what we want to focus on, which is the emissions.

A screenshot of a computer program

Description automatically generated

Here is an image with the LABEL added back to a data frame. The label in this data tells whether the rocket was a mega constellation launch or not. True for yes; False for not. Before we can get into the Decision Trees, let’s look at the balance of the label data right now.

A graph of a bar chart

Description automatically generated

Clearly, the dataset has an imbalance in favor of False. To counteract this disparity, we will over-sample the True data using Imblearn as we have done before. This will give us a balanced data set.

A blue and orange rectangular bars

Description automatically generated

Now we are ready to split our data into training and testing sections for the decision tree learning. Here are pictures of the split data framesA screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

Next, we want to save the labels and remove them from these data frames.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Now that we have our data split and ready for Decision Trees, let’s take a deeper look into the training data.

A diagram of a graph

Description automatically generated with medium confidence

Because of the large number of variables in this dataset—over 20 different quantitative columns—the best chart to look at first is a correlation matrix. A lot of the relationships in this chart are not that surprising. The Stage and Prop Masses for the various stages all having high correlation is self-explanatory. The fact that CO2, CO, and BC are all correlated heavily—was something we explored in the KMeans section of this paper. I believe the most interesting relationship though is how uncorrelated NOx emissions are with every other pollutant. Most of the emission gases have some degree of correlation with another. If the emissions are high that usually means, there’s a mix of all of them in the fuel. However, NOx doesn’t seem to fit that mold as it is weakly negatively correlated with everything. I wonder if this is a result of the fuel type that produces the most NOx and if it is relatively low on the rest of the combustion byproducts.

A graph of a number of objects

Description automatically generated with medium confidence

If we look next at this Kernel Density Estimation graph, the most interesting thing about is, that as CO2 emissions increase, the number of boosters is way more likely to be zero. This is probably the case because booster rockets do not use CO2-emitting heavy fuel types. This is a surprising aspect of the data for me. I would have believed that the environmental impact would have gone up with the number of boosters. In addition, the training data seems to only have one entry for the 6-booster rocket, causing that high peak in the middle of the data. Although I don’t think this will have a strong bearing on the decision tree, this is an example of the limitations of these kinds of graphs. KDE graphs smooth everything out so that it looks continuous but sometimes, in reality, the data’s actual values are more discrete. I think this is something to consider when making these charts we don’t let the simplified visuals mislead us into the nature of the data. On that note, let’s look at the same type of graph but with both a KDE and bars on it.

A graph of a number of objects

Description automatically generated with medium confidence

This is a good representation of how different plots convey their information. From the KDE lines, the information appears more evenly distributed throughout the spectrum of emissions. However, looking at the bar section, reveals that there are specific spots with high concentrations of values while most parts of the x-axis are empty. This conglomeration of data to specific spots is most likely a side effect of the way the emissions data was collected. The data was calculated by using approximate estimations of gases released during combustion and multiplied by the mass of that fuel in the spaceship. This method isn’t exactly wrong, but it does lead to many entries having the same values of emissions over and over because rockets of the same company and build will all have the same standard specs—meaning they all have the exact same emissions. This uniformity is not reality. Every rocket should have its own unique emissions values. But we lack that direct data, so we use our best estimations.

Decision Trees

A diagram of a computer

Description automatically generated with medium confidenceNow we are ready to run the dataset through our Decision Tree algorithm. After setting that up and running it, we get the following tree.

I gave this on another level of depth since there were some many other factors in this dataset that I wanted to give it more layers to classify everything. Now that we have the tree figured out, we run the testing data through the predict method on the Decision Tree to see how well our accuracy was. Here is the Confusion Matrix.

A red squares with white text

Description automatically generated

This Decision tree was by far the least accurate of my project—with 85% accuracy—but that is still a much better than chance level of accuracy. However, there are some surprising features from this tree that I was not expecting. The first among them is how few splits are using the emissions data. Yes, CO2 is the root element, but after that, many of the large splits are done with the sum of mass elements. I would have believed that the emissions data for Mega Constellation and regular launches would have been different enough for the algorithm to focus on those variables. Instead, it was much more balanced in its variable choices than I would have expected. One explanation could be that the emissions were lacking data. Some of the fuel types used in rockets don’t have all the gas in their combustion byproducts. This means that there is a lot of variance in totals across the board with them. What I think this model does show, is that there are still enough discerning factors in the Mega Constellation rockets that they can be identified easily. This could be useful when we want to compare the emission for any new Mega Constellation rocket types that come out to see if they have normal readings for their class.

# Conclusions

One of the main takeaways from this paper is the dominance of Low Earth Orbit (LEO) satellites—a trend that is expected to continue shaping the satellite industry in the coming years. LEO satellites, which orbit much closer to Earth than those in higher orbits, have become essential for services such as global communications, weather forecasting, and Earth observation. Their proximity allows for faster data transmission, lower latency, and more frequent updates, making them ideal for modern applications like internet connectivity and navigation. The growing number of LEO satellites also reflects the increased demand for global connectivity, particularly in remote areas where traditional infrastructure is lacking. However, as more satellites are launched into this already crowded orbit, concerns about space congestion and collision risks are also rising. However, it was shown how satellite companies have attempted to combat this crowding by the increased Mega-Constellation launches. These launches can get many small orbitals in space with a single rocket, significantly lowering the chance of satellite-to-satellite impacts. Despite these challenges, LEO remains the most practical and cost-effective orbit for new satellite deployments, ensuring its continued dominance in the industry.

A major factor driving the growth of satellites is the significant reduction in the cost of launching payloads into space. In the past, the high price of launching even a small satellite made space accessible only to governments and well-funded organizations. However, advancements in reusable rocket technology, increased competition among private space companies, and improvements in launch efficiency have dramatically lowered these costs. This decline in price per kilogram of payload has opened space to new players, including smaller businesses, universities, and developing nations. The result is a more diverse and dynamic space industry, where innovation is thriving and new applications for satellites are emerging rapidly. Lower launch costs also mean that companies can afford to send up larger constellations of satellites, improving coverage for internet services, Earth monitoring, and disaster response. As this trend continues, it can be expected that space will become even more integrated into everyday life, further reinforcing the importance of satellite technology.

While the rapid expansion of satellites brings many benefits, it also raises concerns about environmental impact, particularly in terms of emissions from rocket launches. Each launch releases pollutants into the atmosphere, and with an increasing number of satellites being sent into space, these emissions are becoming a growing issue. However, the rise of mega constellations—large groups of satellites launched together in a single mission—offers a more efficient way to expand satellite coverage while reducing the number of launches. By deploying multiple satellites at once, companies can maximize payload efficiency, leading to fewer overall launches and a smaller environmental footprint. This approach not only makes economic sense but also helps mitigate some of the concerns surrounding space pollution. Nevertheless, as satellite networks grow, industry leaders will need to explore sustainable practices, such as developing greener propulsion systems and designing satellites that can be safely deorbited at the end of their lifespan. Addressing these environmental concerns will be crucial to ensuring that satellite technology remains both beneficial and responsible in the long term.

SpaceX Falcon 9 Booster Reentry

Overall, the future of satellites looks promising, with continued growth, lower costs, and more efficient launch strategies driving progress in the industry. The ability to launch more satellites at a lower price is expanding access to space and fostering new innovations across various fields. At the same time, the industry must balance this rapid growth with responsible environmental practices to ensure sustainability. With careful planning, technological advancements, and international cooperation, satellite technology will continue to shape industries and improve life on Earth. Whether through enhancing global internet access, supporting scientific research, or aiding in disaster response, satellites will remain a cornerstone of modern infrastructure. The challenge will be managing space congestion and environmental impact while continuing to push the boundaries of what satellites can achieve. If these challenges are met with thoughtful solutions, the future of space exploration and satellite deployment will be both exciting and sustainable.