IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cameron Schlonski
11/23/2021

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- In this project, I used the SpaceX REST API and Wikipedia to get SpaceX booster launch information. Using this information, I wanted to know if I could predict if a booster could land successfully after launching. Using Pandas and NumPy, I analyzed and cleaned the data. Then I used Sci-kit Learn to apply their machine learning models to predict off the data.
- In the end, I was able to create a successful decision tree model that can predict with 83% accuracy. I also learned about the data and was able to share this information visually.

Introduction

- In this project, I intend to find the way to predict the outcome of a SpaceX booster landing. I will gather data from SpaceX, analyze, visualize, and clean the data, and finally apply machine learning models to it and find a best model.
- In this project, I hope to learn about different SpaceX boosters and determine which boosters have better success rates. I also want to know if there is any hidden detail that may have hidden consequences like if newer boosters have significantly higher success rates than older ones.

Section 1

Methodology

Methodology

Executive Summary

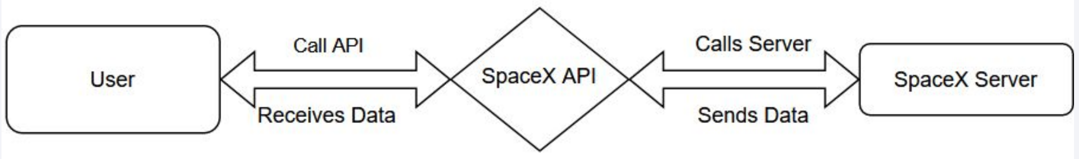
- Data collection methodology:
  - The data for this project was collected using the SpaceX REST API as well as web scraping Wikipedia for extra information.
- Perform data wrangling
  - The data was then processed using BeautifulSoup and Pandas to put the data in a usable format.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - A Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree algorithm was applied to the data. These models were used and then the model with the highest accuracy on the testing data was selected.

Data Collection

- The data sets were collected using the SpaceX REST API and web scraping Wikipedia tables using BeautifulSoup.
- Using the API, I was able to connect to the SpaceX server and gather information about their past booster launches. This data was then passed onto a Pandas DataFrame in order to visualize and analyze the data.
- I also used BeautifulSoup, a web scraping library in Python, to gather information from a Wikipedia table on SpaceX launches. This data was then processed and passed along using a Python dictionary and then passed on to Pandas.
- These two datasets were combined using Pandas to create a combined set for the whole project.

Data Collection – SpaceX API

- In this step, I accessed the SpaceX REST API in Python in order to gather their data and make it usable in a Pandas DataFrame for analysis.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20Data%20Collection%20w/API.ipynb



Data Collection - Scraping

- Utilizing BeautifulSoup, I web scraped the Wikipedia page to gather more data about the SpaceX launches.
- I web scraped information about the SpaceX Falcon 9 and Falcon Heavy launches from the Wikipedia table. In the data wrangling step, I simplified this down to just the Falcon 9 launches though.
- The extra information that came from this included info on: Flight Number, Launch Site, Date, Time, the Payload, the Payload Mass, the Orbit Type, the Customer, the Launch Outcome, the Version Booster, and if there was an attempted Booster Landing.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20Data%20Collection%20w/webscraping.ipynb

Data Wrangling

- Once I obtained the data through the API and scraping, I needed to understand the data and each of the features.
- Each of the features were important, but there certain features that stood out and I wanted to look more into such as the Launch Site, the Orbit, and the Launch Outcome.
- Each of these features likely hold important information that can be used within the predictive analysis part.
- I also created an Outcome feature to have an easy to read feature containing the info of whether or not the launch was successful.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Visualizing the data helps to see patterns or hidden aspects of the data.
- Here I visualized the relationships between the Flight Number and Launch Site, Payload Mass and Launch Site, the Orbit and Class, Flight Number and Orbit, Payload Mass and Orbit, and the success rate over time.
- Some key takeaways:
    - The flight number had a positive relationship with the Class for each launch site.
    - The higher the payload mass, the higher the success rate for all launch sites.
    - The higher the orbit, the higher the success rate from the bar chart. However, the earlier flight numbers were usually the lower orbits and the earliest trials. As they got more successful, they increased the orbit size which is why the higher orbits are more successful.
    - Payload mass is a strong indicator for certain lower orbits but not others.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20EDA%20w/visualization.ipynb

EDA with SQL

- Utilizing SQL, I was able to analyze the dataset.
- There are 4 distinct launch sites for SpaceX,
- The first successful mission was in 2010, but the success rate has improved drastically since then.
- Since the start, there has only been 1 unsuccessful mission outcome and 100 successful mission outcomes.
- There is only 1 booster, the F9 B5 that carries the max payload mass.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20EDA%20w/SQL.ipynb

Build an Interactive Map with Folium

- In my project, I also utilized Folium to visualize the USA and the launch sites.
- The launch sites are located in either California or Florida.
- I added a mouse position object to gather coordinate data where the mouse is located. This way I can calculate distance and lines between the launch sites and other structures.
- I was able to create lines between the Florida launch sites and the coast as well as the nearest highway.
- This showed me the ease of access to these launch sites as well as that they are separated from people for protection. They have access to their own roads for transporting the rockets as well as easy access to the water in order to recover successful and unsuccessful landings.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20Dashboard.ipynb

Build a Dashboard with Plotly Dash

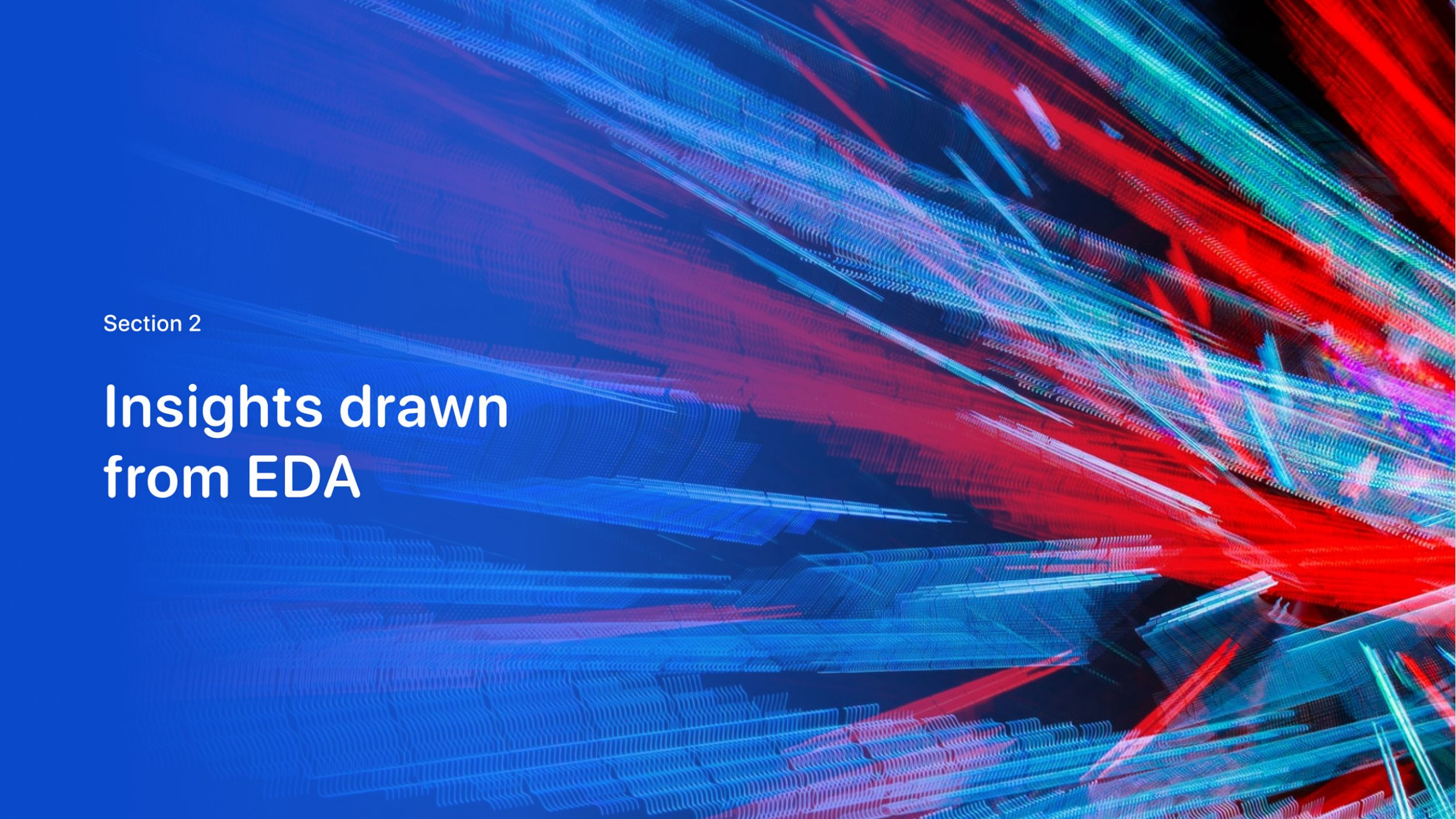
- Using Plotly and Dash, I constructed an interactive dashboard to represent the launch data in order for others to gain an understanding of the data.
- The pie chart helps to represent the total number of launches and where they came from as well as successful and unsuccessful launches per site.
- The scatter plot shows the successful and unsuccessful launches by payload mass for each launch site.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20Plotly%20and%20Dash.ipynb

Predictive Analysis (Classification)

- Utilizing Sci-kit Learn, I fit different models to the data and predicted off of the test set to determine which model would best fit the data.
- After already cleaning and standardizing the data, I applied the train_test_split function to the data.
- Then I applied a KNN, SVM, Logistic Regression, and Decision Tree model to the training sets. From here, in order to optimize, I used GridSearchCV from Sci-kit Learn. This optimized the hyperparameters for the models and gave a best model for each.
- Then I applied the models to the test set, only to realize that they all had the same accuracy on the test set. That is when I used their training set accuracies to determine that the Decision Tree model would be best for the data.
- https://github.com/cschlons/First-DS-Project/blob/master/Capstone:%20Plotly%20and%20Dash.py

Results

- After exploring the data, I found out that the orbit size matters in regards to a successful landing. But when payload size and flight number are introduced, we see that the earlier flights were more likely to be lower orbits. Once they gained more experience and more success, then SpaceX went to higher orbits and higher payload masses.
- I also found that payload mass, in general, was an indicator for success for all launch sites. That is, the higher the payload mass, the higher the chance of success for all launch sites.
- After analyzing the data, I applied the four machine learning models to the data. This led to all models having the same test set accuracy, but different training set accuracies. Based off of this, I believe that the model with the highest training accuracy, the decision tree, is the best model for predicting the landing outcome.

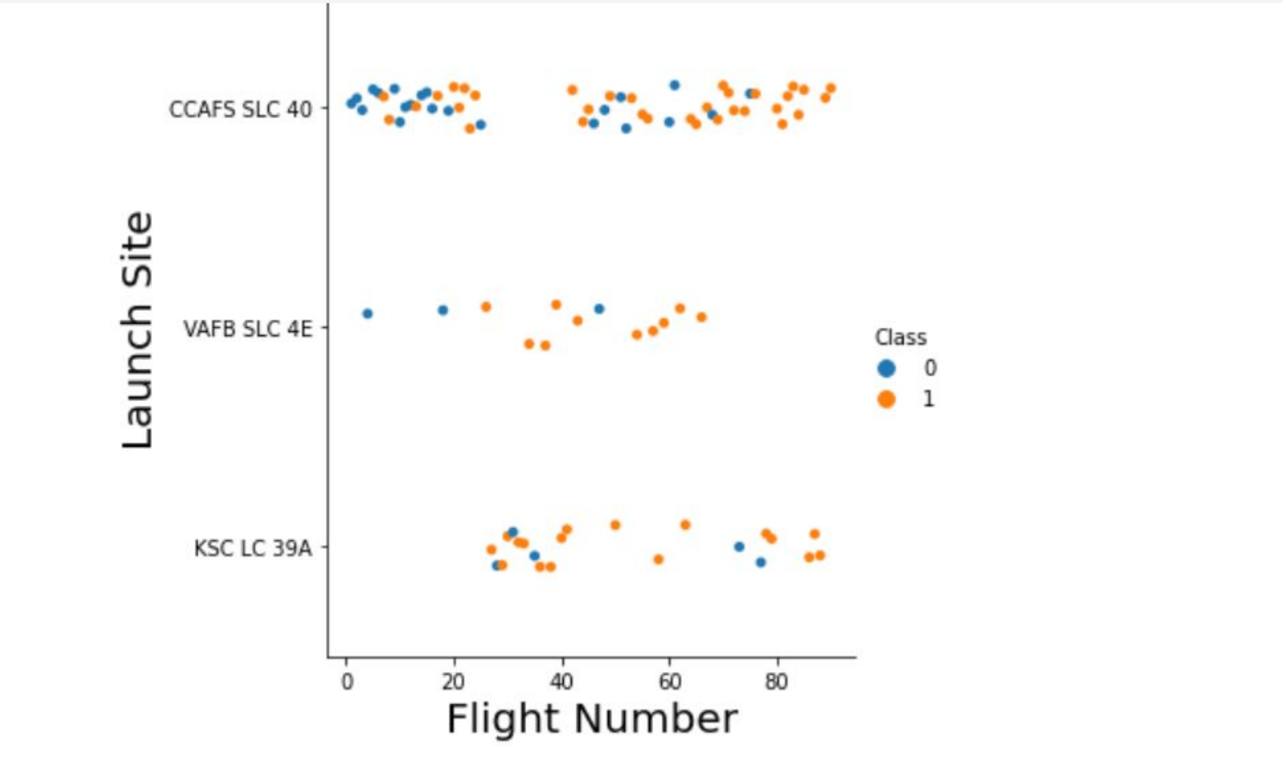


Section 2

Insights drawn from EDA

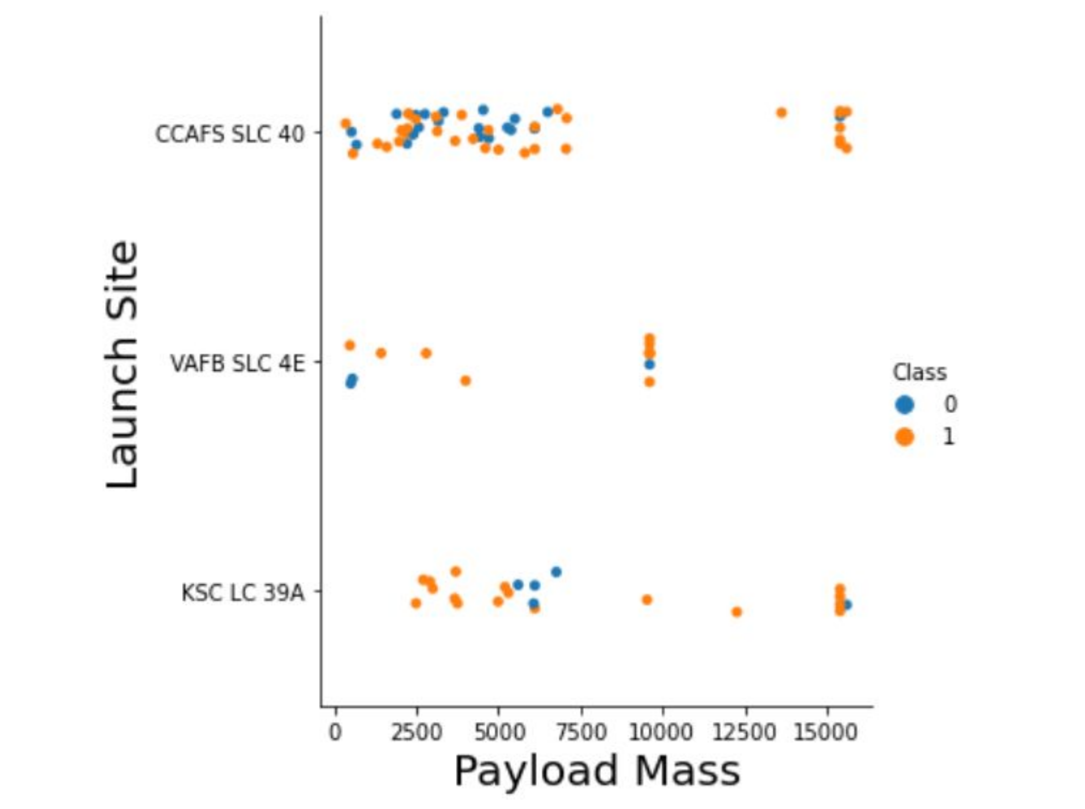
Flight Number vs. Launch Site

- This plot depicts the relationship between the launch site and the flight number.
- The higher the flight number, the more likely the class is 1 for all of the launch sites.
- CCAFS has the most launches and VAFB has the least number of launches, but they still have high success rates.



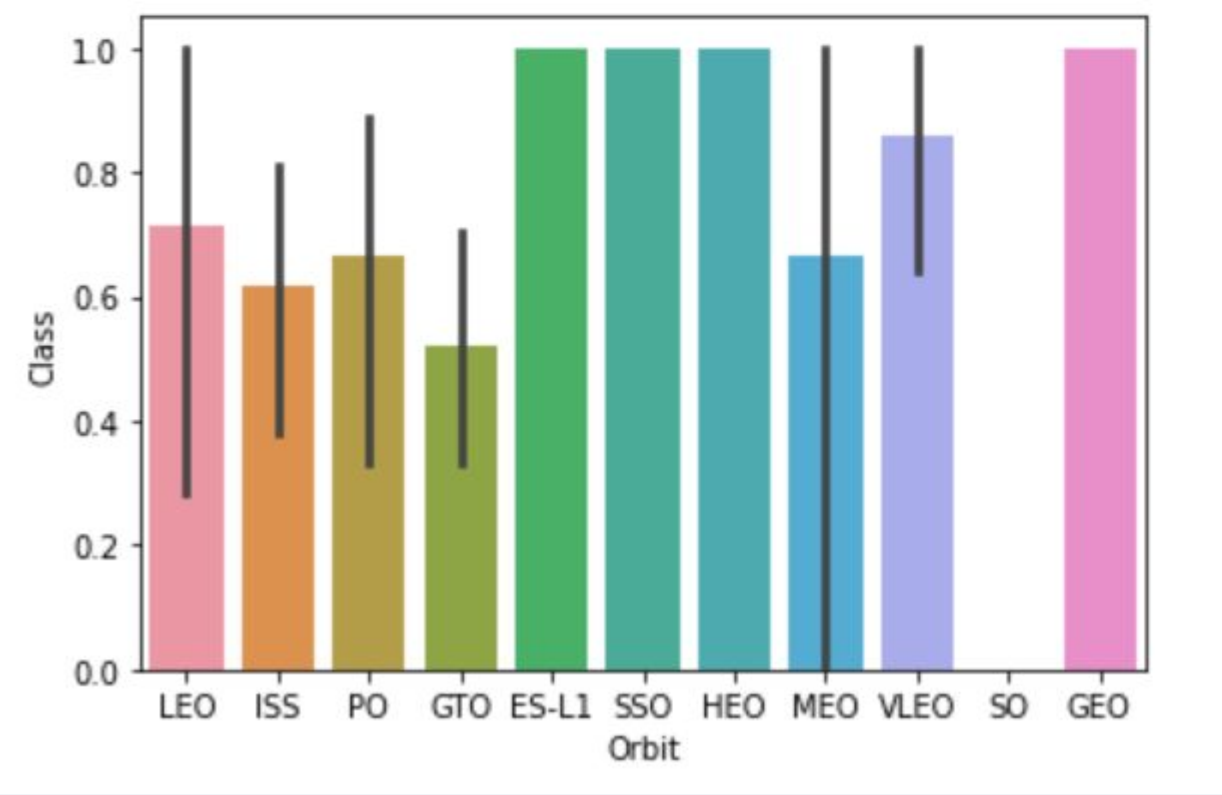
Payload vs. Launch Site

- This plot depicts the relationship between the launch site and the payload mass.
- The higher the payload mass, the higher the success rate for all launch sites.
- VAFB doesn't go above a certain payload mass.
- CCAFS launches most masses except for the range of 7500-12500.



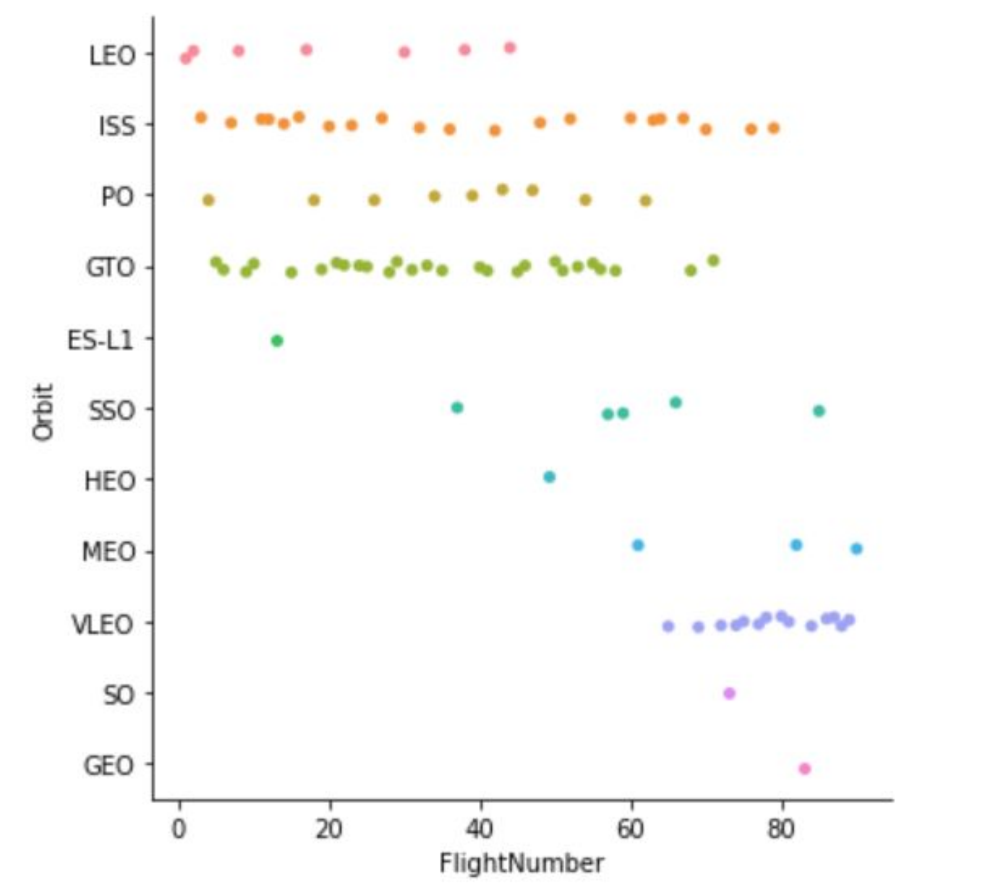
Success Rate vs. Orbit Type

- This plot depicts the success rate of launches for each orbit.
- The higher the orbit, the higher the success rate.
- It appears that there is a large standard deviation for the lower orbits and none for the higher orbits. This may be because the lower orbits have a higher sample size than the higher orbits.



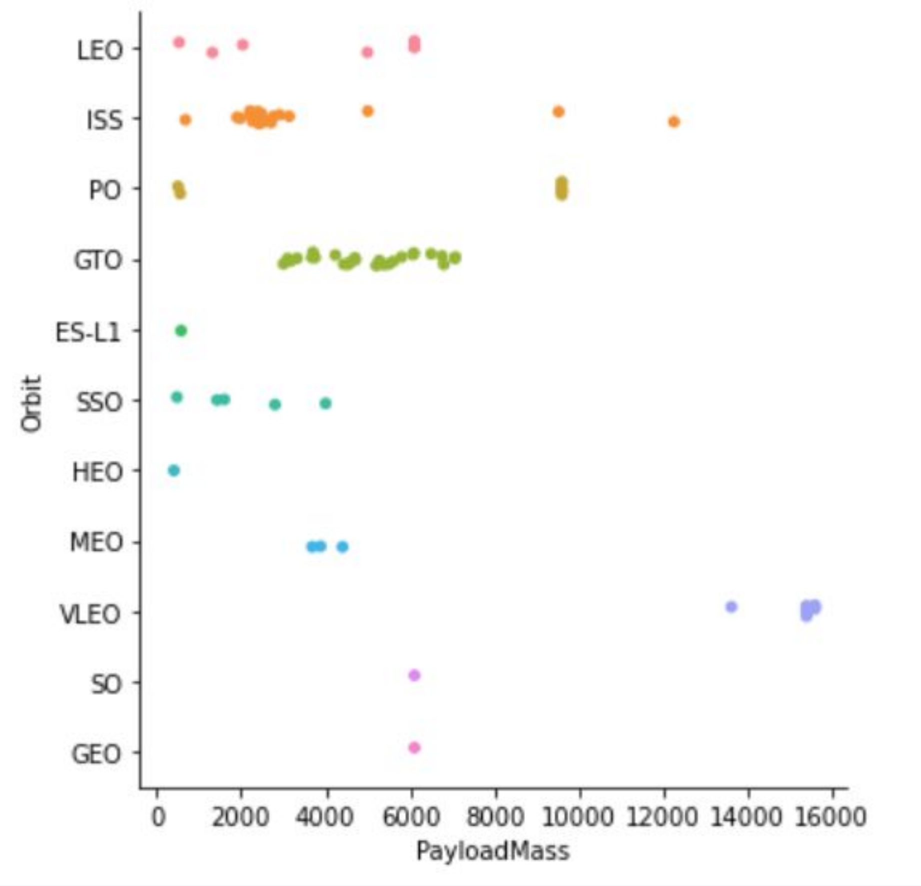
Flight Number vs. Orbit Type

- This plot depicts the relationship between the flight number and the orbit.
- It appears that the higher the orbit, the higher the flight number.
- This might relate to the previous slide. Since the lower flight numbers correspond to the lower orbits, then that might be why the lower orbits have a lower success rate. Because SpaceX was newer to this and was performing more trial and error before they had a better idea on how to launch the payloads.



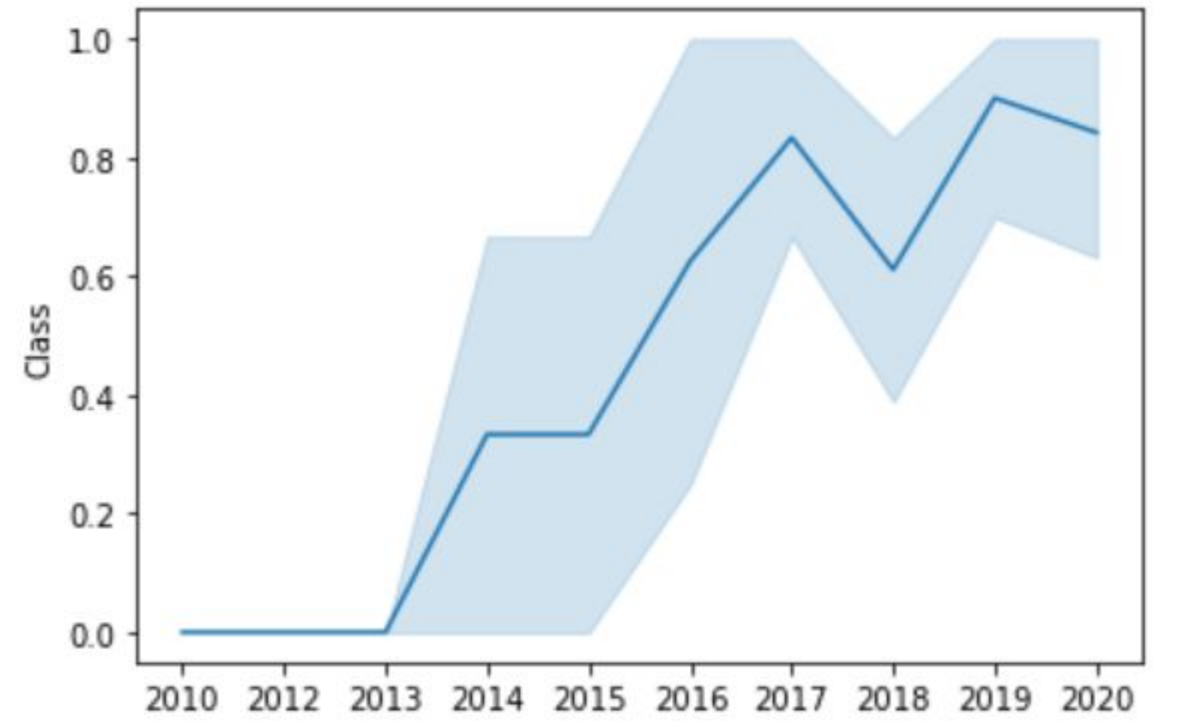
Payload vs. Orbit Type

- This plot depicts the relationship between the payload mass and the orbit.
- It appears that lower payload masses are more likely to be flown into a lower orbit.
- Whereas we cannot be certain that higher payload masses are indicative of higher orbits as can be shown by the ISS and PO.



Launch Success Yearly Trend

- This plot depicts the success rate by year.
- This plot clearly shows growth in success rate after 2013 with a near positive increase every year except for 2014, 2018, and 2020.
- Although there are some decreases in success rate, the rate is still quite high and near and above 80% since 2017.



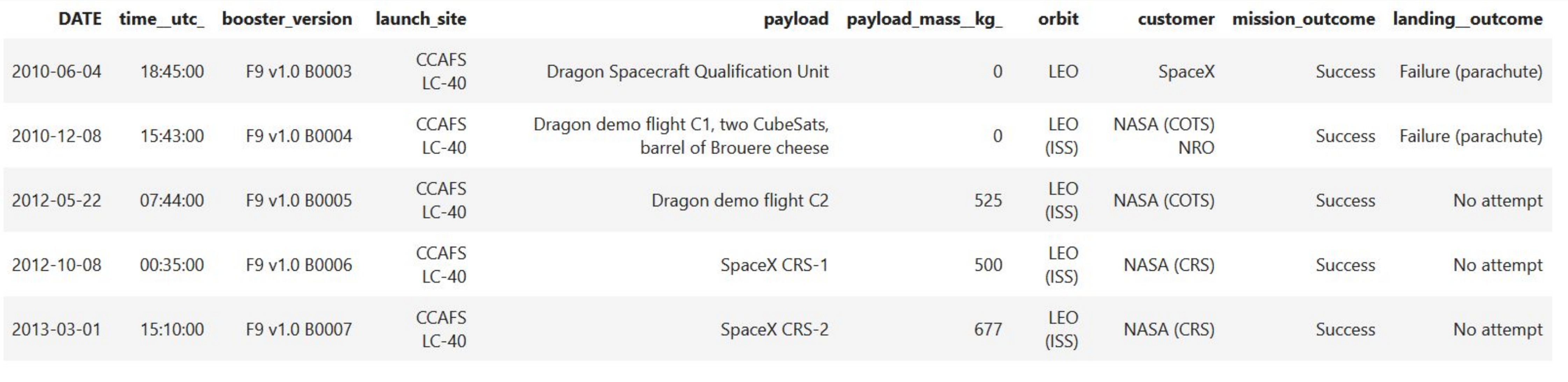
All Launch Site Names

- The names of all launch sites:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E
- This query looks for each distinct launch site in the SpaceX table.

```
SELECT DISTINCT(Launch_Site) FROM SPACEXTBL
```

Launch Site Names Begin with 'CCA'

- This query selects 5 rows of the table where the launch site begins with CCA.
- This launch site seems to contain the first attempts at launching the payloads.
- According to SpaceX, they all are successes with failed or no attempts as landing.



DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload mass carried by boosters launched by NASA is 48213 kg.
- This query summed up all the payload masses for NASA launches.
- The mass seems to be very high, which could be due to carrying ISS equipment or additions to space.
- It may also be that since this mass is very high, that NASA uses SpaceX for the majority of taking their payloads into space.

Average Payload Mass by F9 v1.1

- The average payload mass for the F9 v1.1 booster is 2534 kg.
- This is an average or lower mass for a payload.
- It is possible that since this is a lower version booster, that it was used for building up to and training for heavier payloads.

First Successful Ground Landing Date

- The first successful ground landing date was on June 4, 2010.
- The missions go back to 2010, and it seems that their first successful ground landing was very early on in their trials.
- Due to their early success, they could have gathered the attention NASA and could be the reason why they had many launches to bring payloads to the ISS.
- Their early success could also be attributed to why they have such high success rates for more recent launches.

Successful Drone Ship Landing with Payload between 4000 and 6000

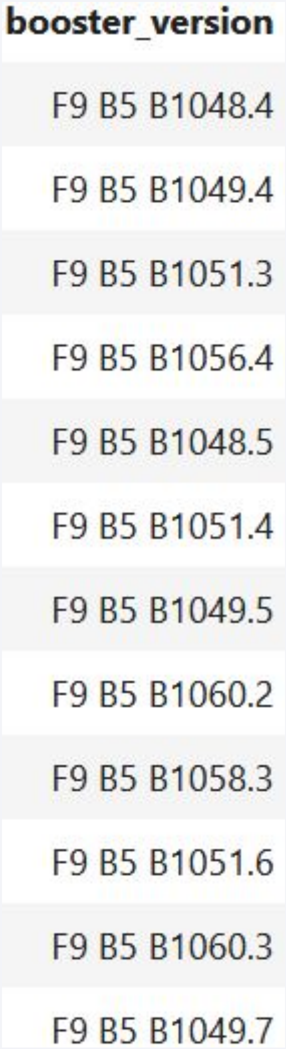
- The boosters that successfully landed on the drone ship with payload masses between 4000 and 6000 kg are:
  - F9 FT B1020
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2
- There are few successful landings with this payload mass. It may be that this payload mass requires too large of a booster to land on a drone ship.
- It could also be that their more recent models for the boosters are better at landing on the drone ship while also carrying a larger payload.

Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes is 100 and failures is 1.
- It is important to point out that successful doesn’t necessarily mean landing.
- In the early stages around 2010, they unsuccessfully landed boosters and even didn’t attempt to land boosters. These mission outcomes were considered a success.
- The mission is important, but the landing outcome is a priority as well.

Boosters Carried Maximum Payload

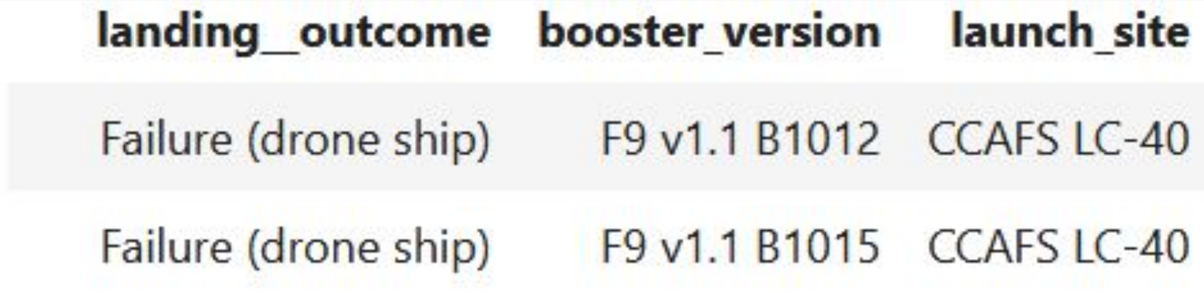
- These are the booster versions that carried the maximum payload mass.
- Comparing the booster version to the earlier ones, it seems that the B5 boosters are designed to carry heavier payloads.
- It also seems that the ending of the version, i.e. B1048.4, is like a version number. This could mean that the 4th trial of the 48th booster was successful in carrying the largest payload mass.



booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

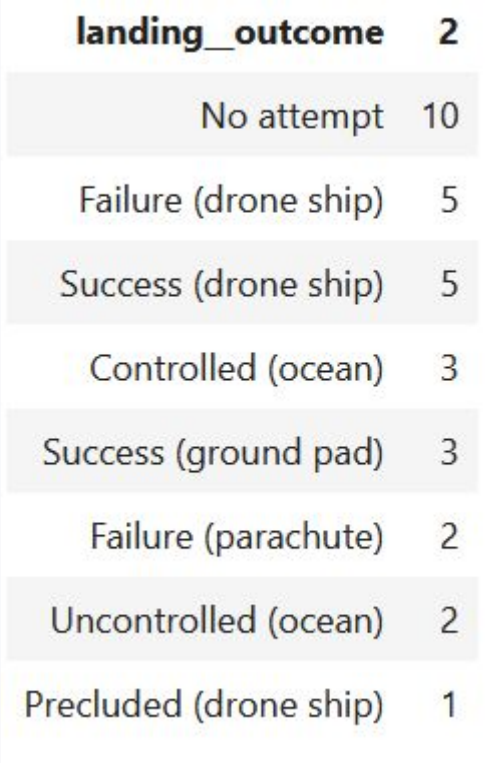
- This list shows the outcome of the failed landing outcomes for attempts on a drone ship in the year 2015.
- These failures were done with what appears to be an earlier version booster, with both being launched at the same site.



landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- This table shows the rank for the outcome of all launches from 2010-06-04 to 2017-03-02.
- It seems that they were unprepared for attempting landing with the boosters that they were using which is why No Attempt is the highest.
- It also appears that they had an equal number of successes and failures on the drone ship. It is possible that they were trying newer boosters in this time frame which could account for more failures.



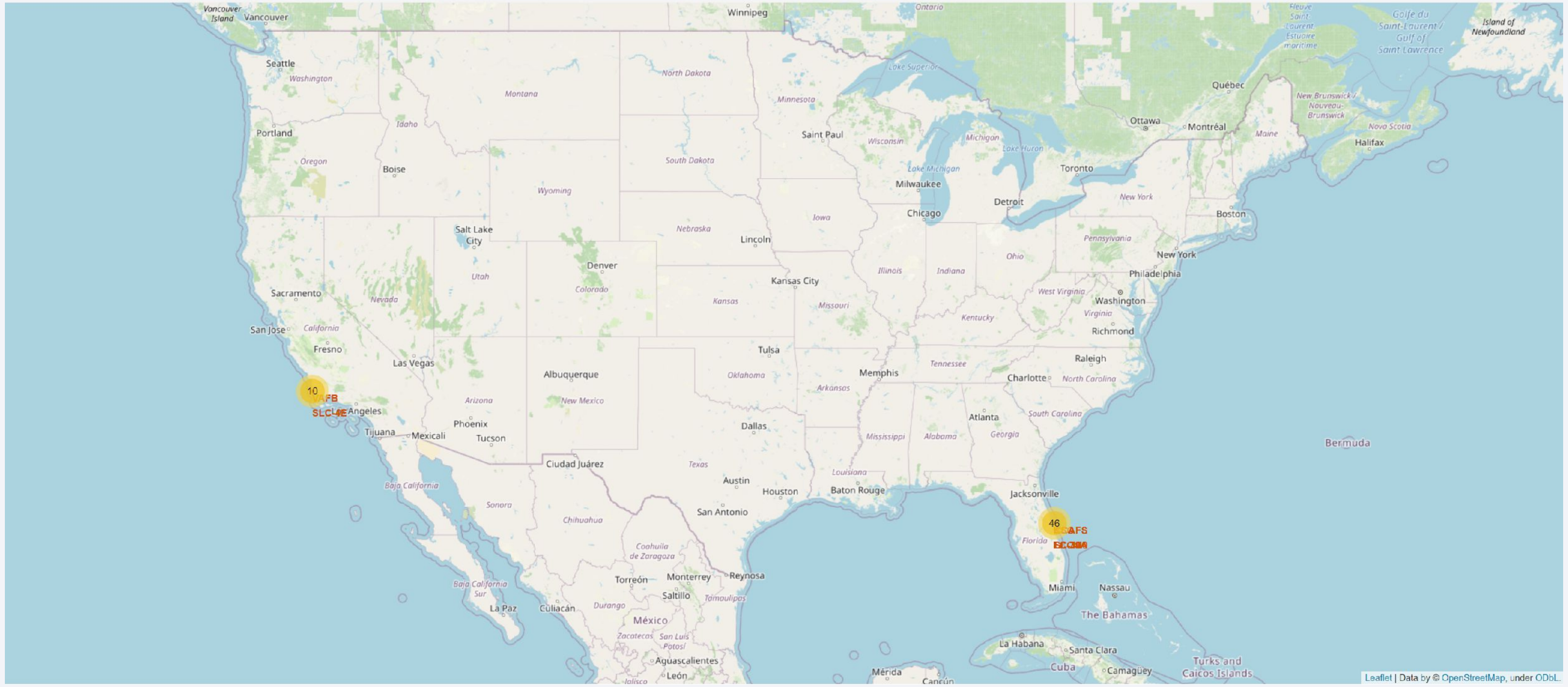
landing_outcome	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

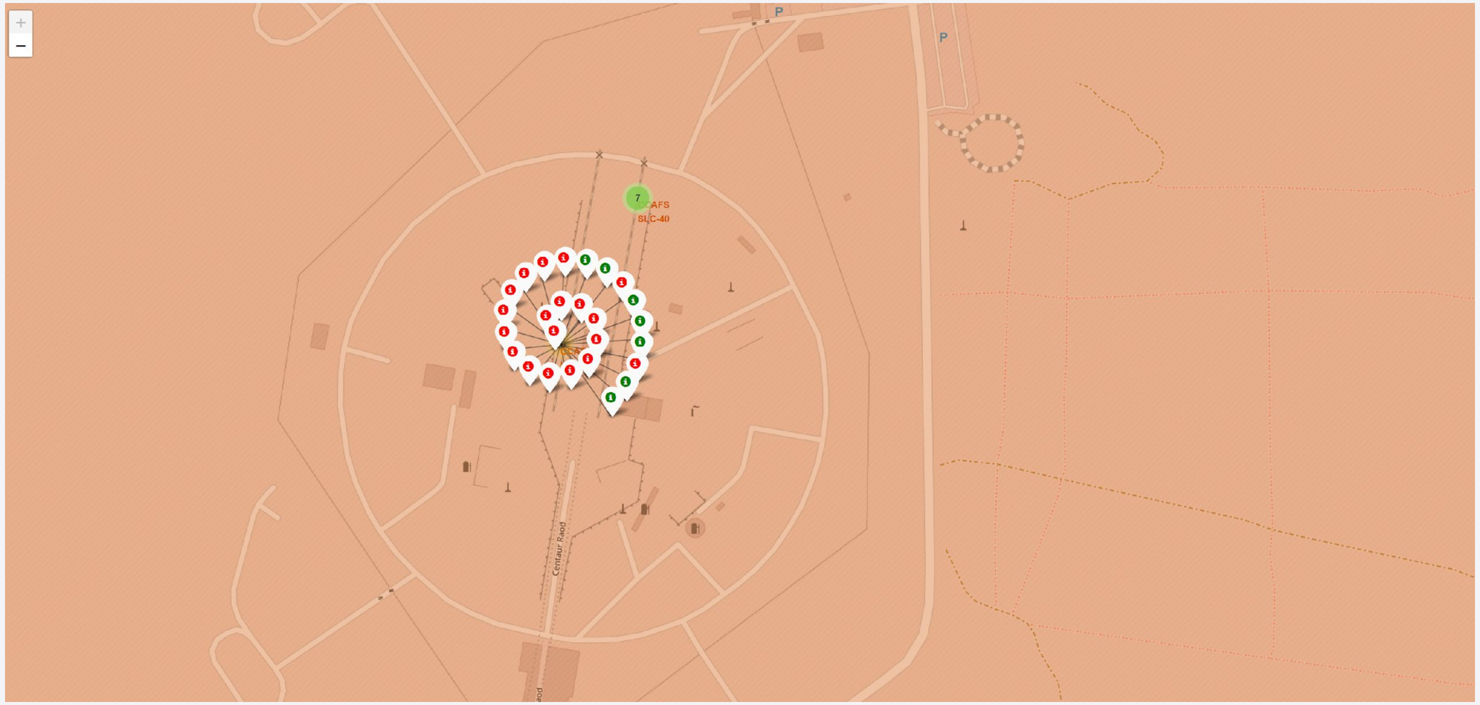
All Launch Sites in the USA

- In this map, you can see that the launch sites are located on the coasts. Since a key part of SpaceX is landing the rockets, this gives them easy access to the water for water landings as well as landing on a drone ship.



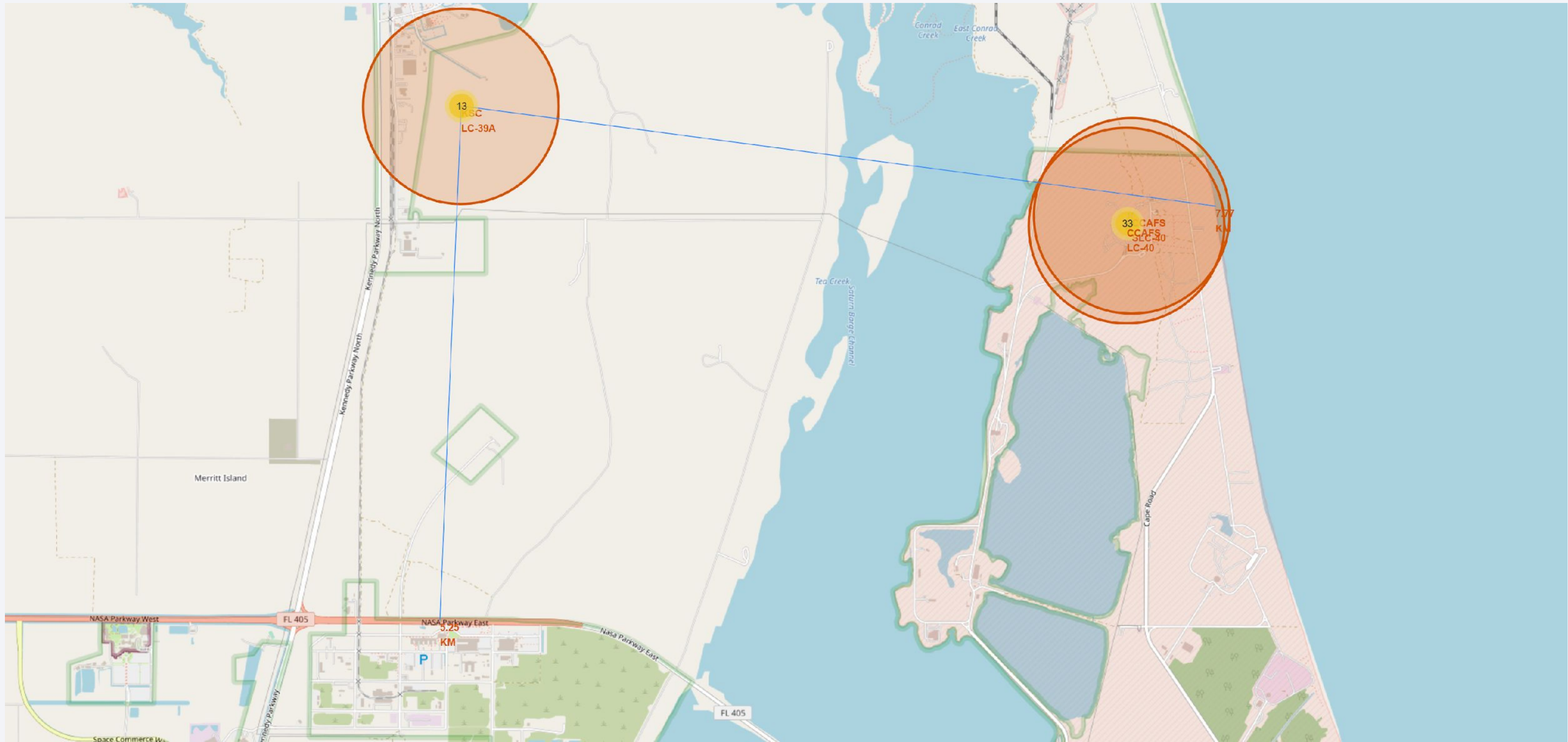
Launch Sites with Outcomes Labelled

- These sites show green and red markers. These markers show the successes and failures with green and red colors respectively. This information can help us see which launch sites are used more often and whether or not they have successful launches.



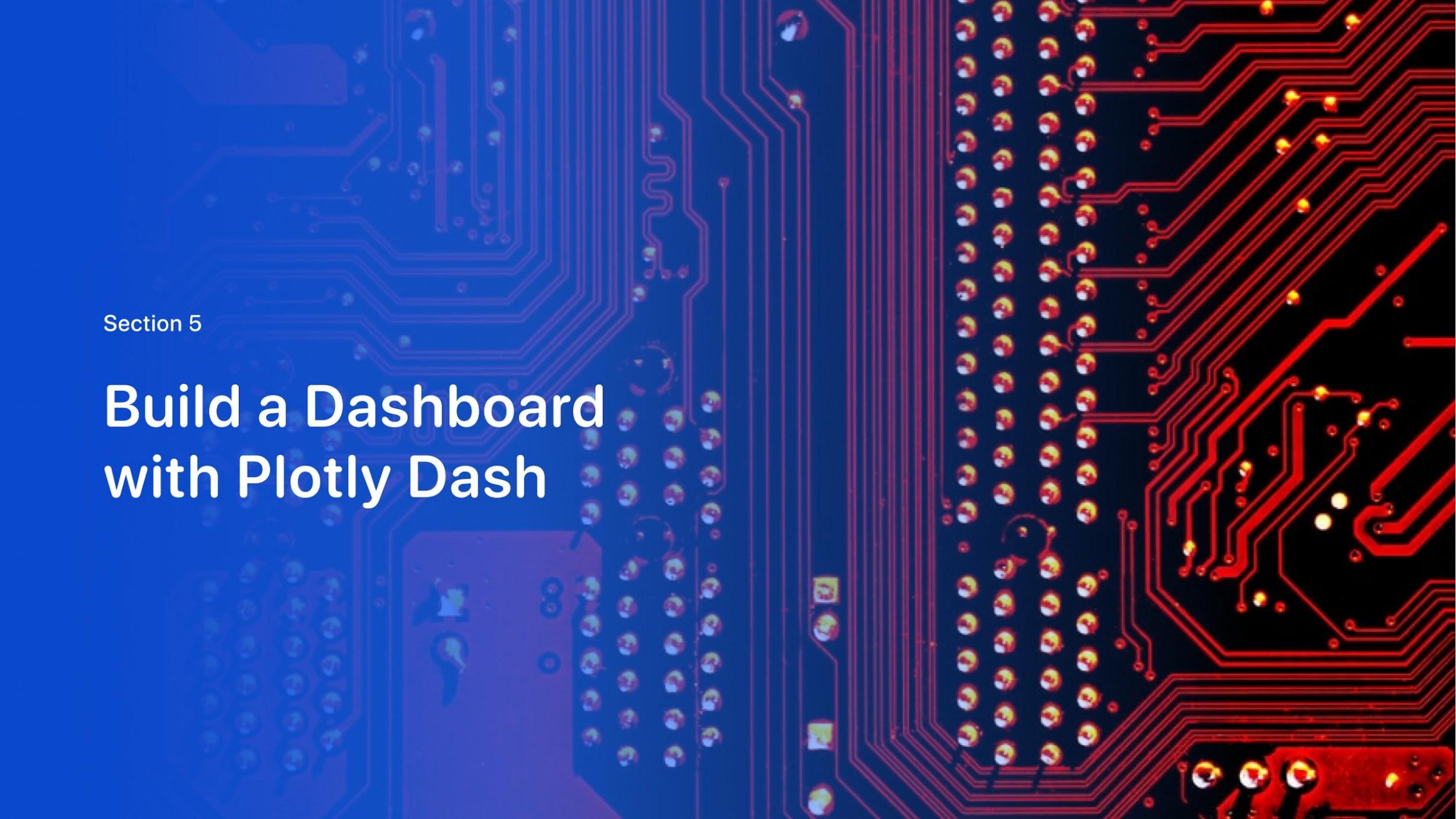
Launch Site Distance to Nearby Locations

- In this image, we can see that the KSC LC-39A site is in a similar location to the CCAFS launch site. The difference, however, is that the KSC site is father from the coast, but close to the nearby highway. This gives easier access to supplies, but provides a difficult problem of retrieving the boosters after a successful landing.



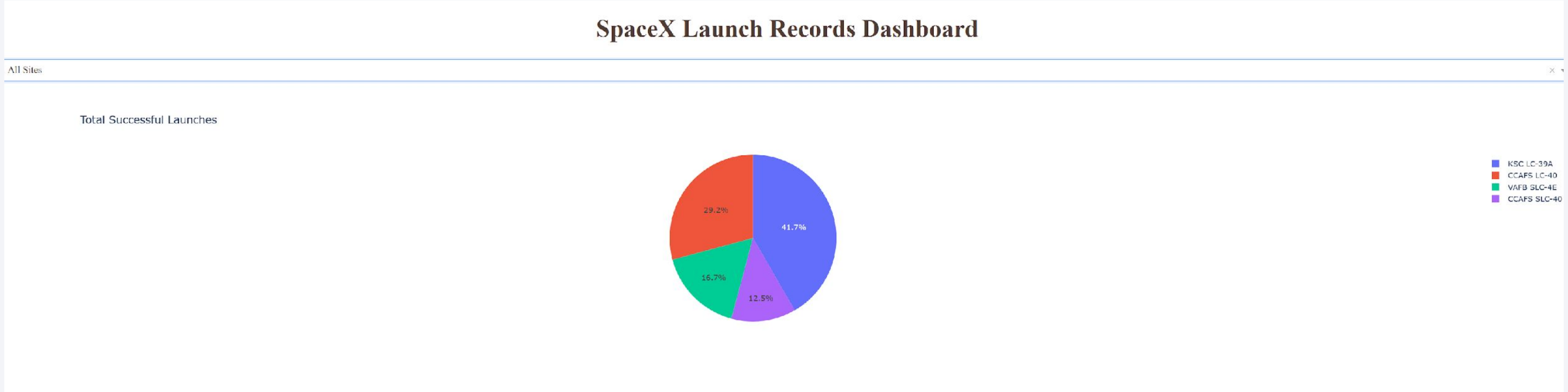
Section 5

Build a Dashboard with Plotly Dash



Total Successful Launch Count (All Sites)

- According to this pie chart, across all launch sites, KSC LC-39A has the largest number of successful launches.
- This tells us that this launch site would be most beneficial to try new boosters due to the high success rate.
- This success rate could be attributed to a preferable location, or that they test out boosters that are already known to have a high success rate.



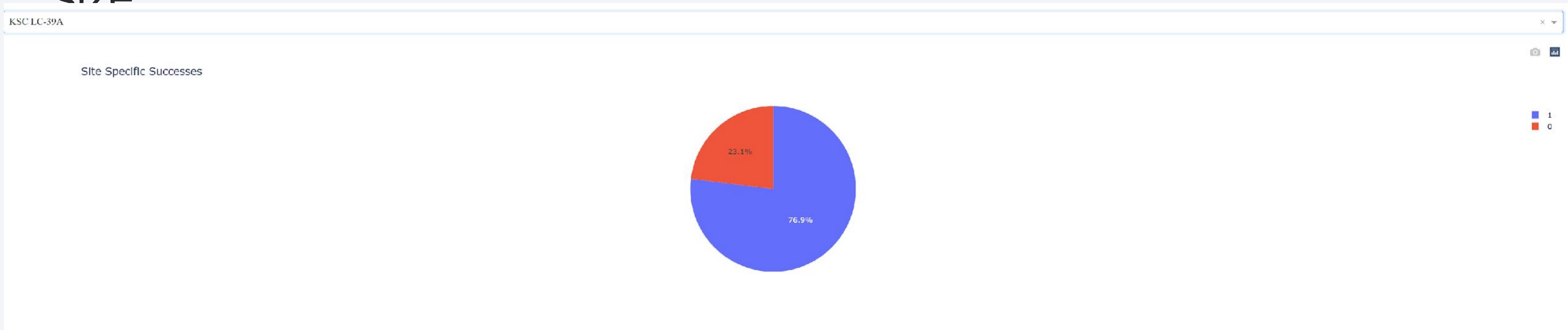
SpaceX Launch Records Dashboard

All Sites

Total Successful Launches

Highest Success Ratio Launch Site

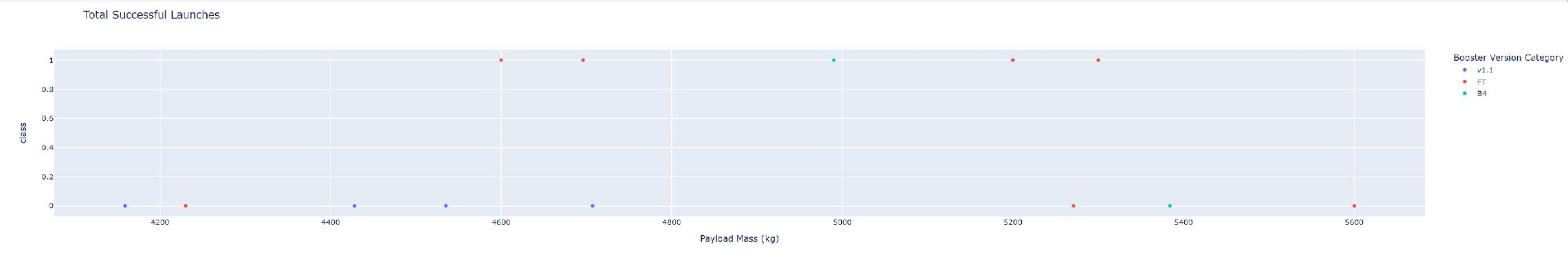
- According to this pie chart, KSC LC-39A is the launch site with the highest success ratio for launches.
- This tells us that if we want to launch at a site with high certainty of success, we should choose KSC LC-39A.
- Even though this site is not, the most used site, it is still the second most used site. That means that its high success ratio is likely not attributed to a low sample size



KSC LC-39A

Payload Size vs Launch Outcome for all sites (4000-6000 kg)

- According to this scatter plot, within the payload range of 4000 to 6000 kg, the v1.1 booster type is more successful than the B4 booster type.
- Although the B4 booster type is used on the lower range of this scale, it has a 100% failure in this mass range.
- Whereas the v1.1 booster has the highest success rate of any booster in this mass range for all sites.

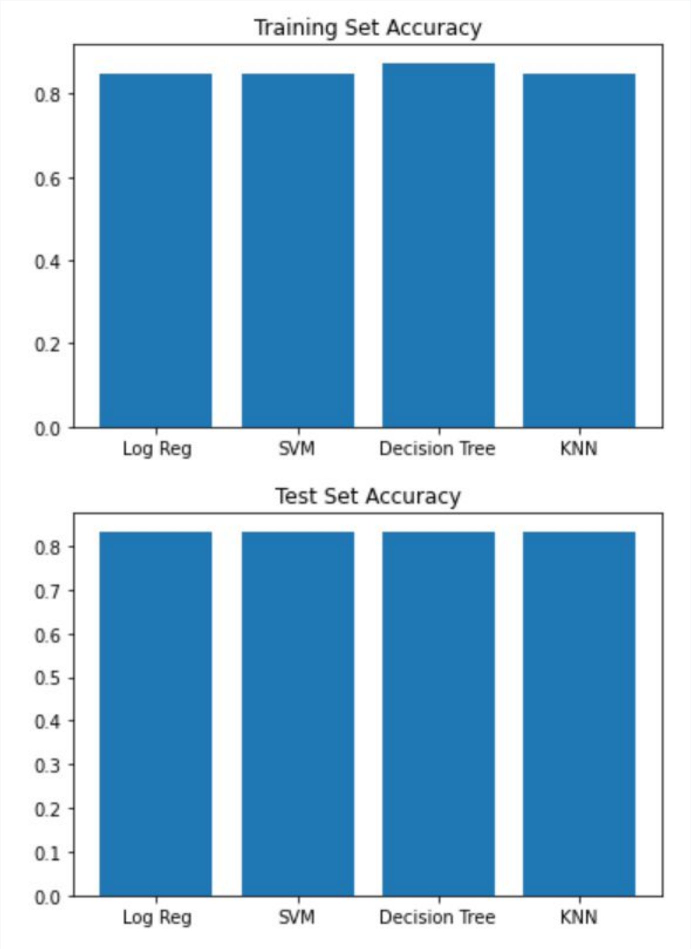


Section 6

Predictive Analysis (Classification)

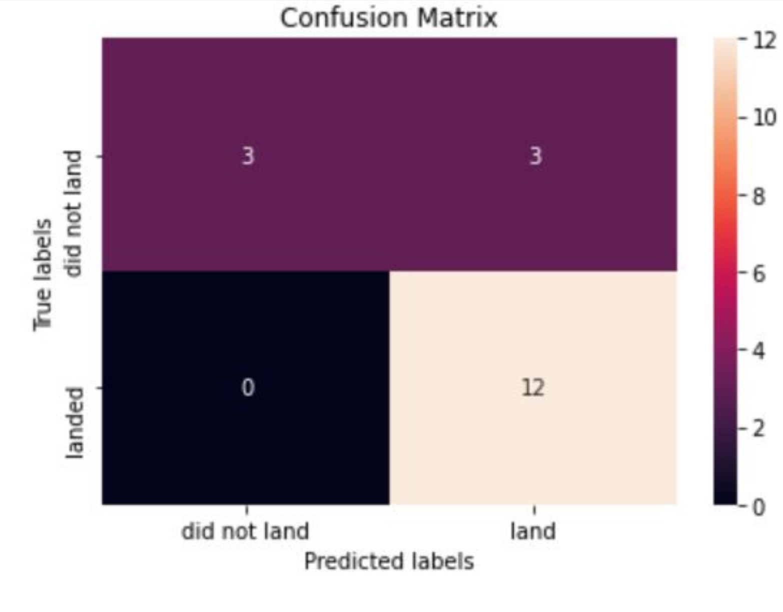
Classification Accuracy

- After applying each model to the training data, we can see that the decision tree had the best accuracy around 87.5%.
- After applying each model to the test data, we can see that all models had the same accuracy with 83.3%.
- Therefore, it is reasonable to choose the decision tree model due to have the best training accuracy and equal test accuracy.



Confusion Matrix

- According to the confusion matrix for the decision tree, it correctly labelled every unsuccessful landing and labelled 12/15 successful landings.
- This tells us that the model is very good at determining true unsuccessful landings, and moderately good at predicting successful landings.

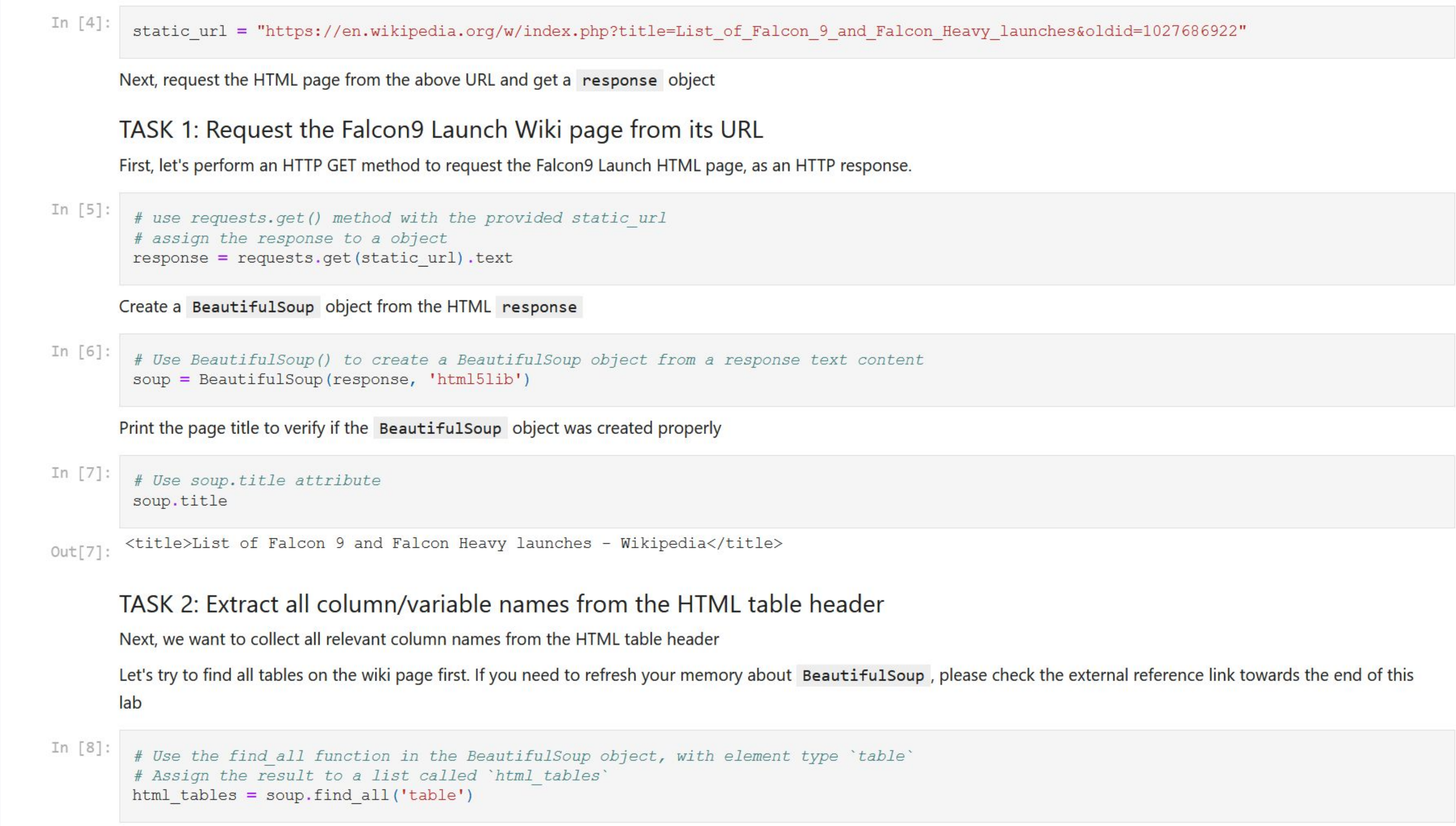


Conclusions

- Using SpaceX's REST API and webscraping Wikipedia, I was able to gather a large sample and accurate data of their launch history.
- After analyzing the data, I was able to determine that each launch site had it's own success rates and that they needed to be one-hot encoded for model training.
- Using visual analysis, I was able to graph and compare different features in order better understand their impact on the problem.
- Using Sci-kit Learn, I was able to clean the data and engineer new features for training of different models.
- Using the final training and test data sets, I was able to determine that the Decision Tree best fit the data and had the highest training and test accuracies.

Appendix

The web scraping code and the Wikipedia link used.



```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        response = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(response, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]: # Use soup.title attribute
        soup.title

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

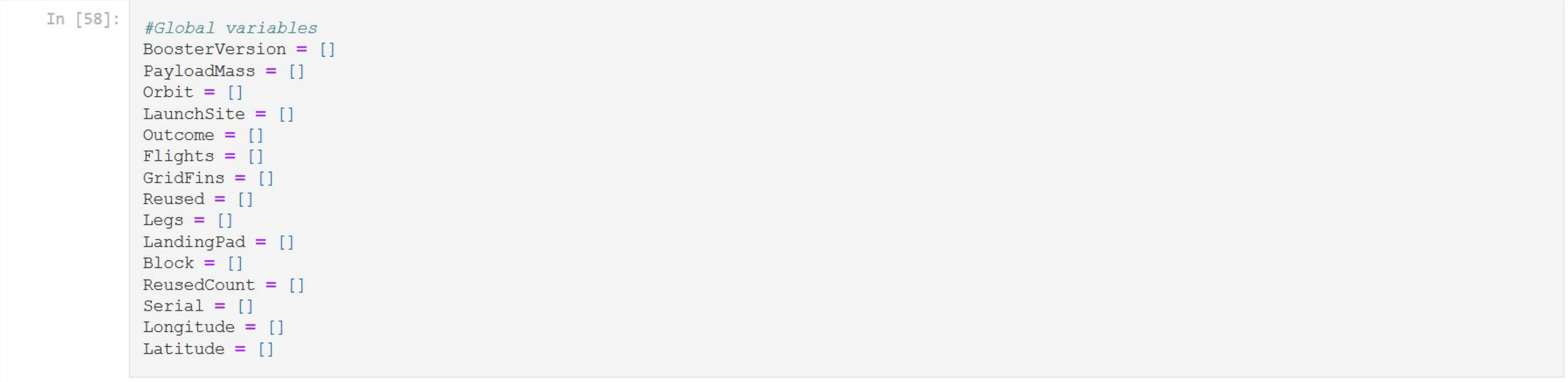
Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
In [8]: # Use the find_all function in the BeautifulSoup object, with element type `table`
        # Assign the result to a list called `html_tables`
        html_tables = soup.find_all('table')
```

Appendix

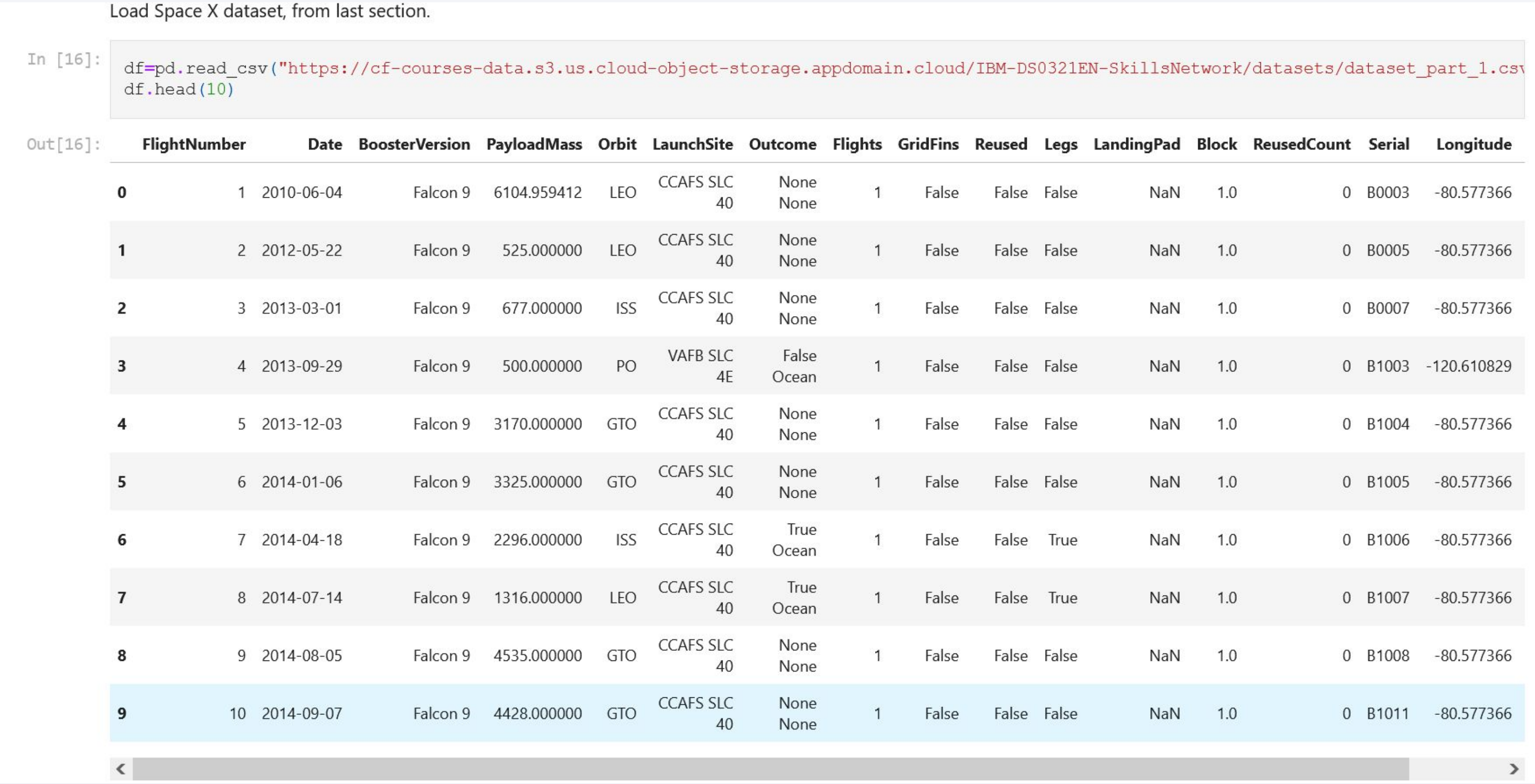
The features gathered from the SpaceX REST API.



```
In [58]:
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

Appendix

The final DataFrame just before the data analysis and feature engineering step.



Load Space X dataset, from last section.

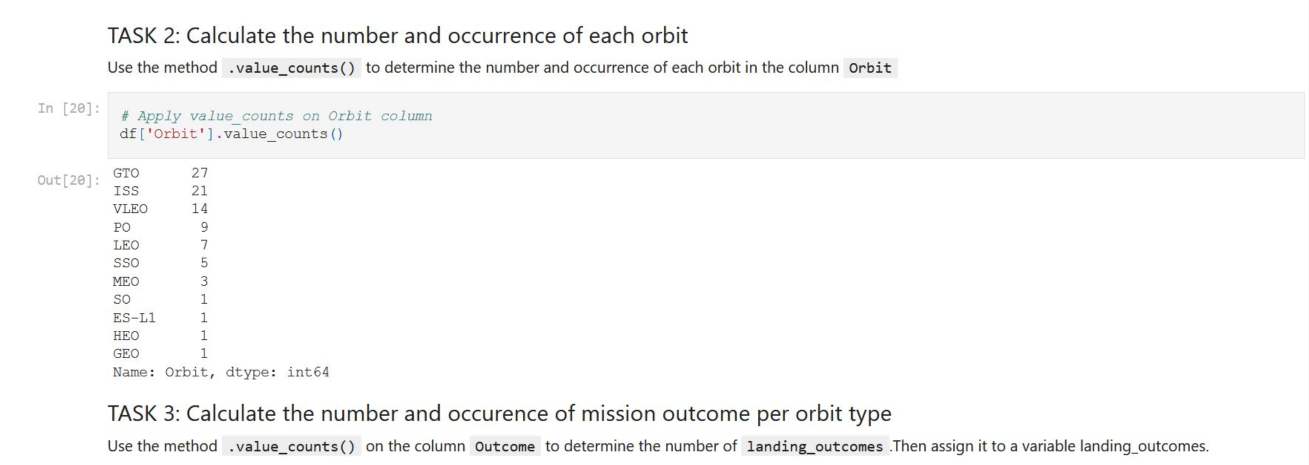
```
In [16]:  df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv
          df.head(10)
```

Out[16]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1005	-80.577366
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1006	-80.577366
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1007	-80.577366
8	9	2014-08-05	Falcon 9	4535.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1008	-80.577366
9	10	2014-09-07	Falcon 9	4428.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1011	-80.577366

Appendix

The orbit information and landing outcome from the data set.



TASK 2: Calculate the number and occurrence of each orbit

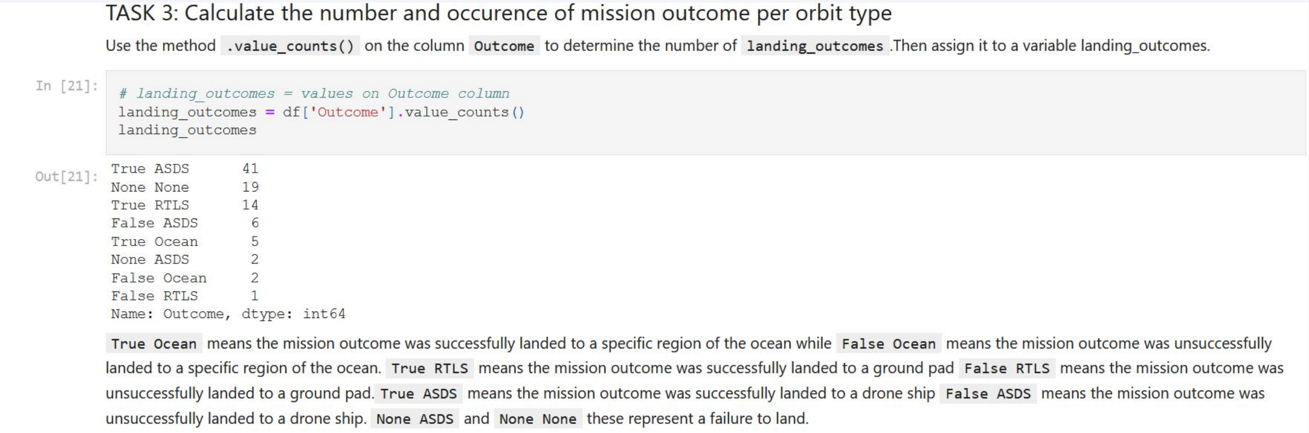
Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [20]:
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Out[20]:
```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome per orbit type

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.



TASK 3: Calculate the number and occurence of mission outcome per orbit type

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

In [21]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

Out[21]:
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

`True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean. `True RTLS` means the mission outcome was successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad. `True ASDS` means the mission outcome was successfully landed to a drone ship `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

Thank you!