IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pierre E Martin
25 March 2024

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data Collection with API
  - ➢ Data Collection with scrapping
  - ➢ Data Wrangling
  - ➢ Exploratory Data Analysis with Data Visualization
  - ➢ Exploratory Data Analysis with SQL
  - ➢ Interactive Visual with Folium and Plotly Dash
  - ➢ Predictive analysis
- Summary of all results
  - ➢ Exploratory Data Analysis result
  - ➢ Interactive Data Analysis Results
  - ➢ Predictive Analysis Results

Introduction

- Project background and context

SpaceX company has a mission Falcon9 that sends rockets to space and ISS. They cost $62million, others cost about $165million; the reason for savings is because SpaceX can reuse the first stage. Therefore if we determine the first stage will land, we can determine the cost of launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The goal of the entire project is to make use of Data Science and build a machine learning model that can predict the first stage will land successfully.

- Problems you want to find answers

⍰ What factors determine the rocket will land successfully?
⍰ The interaction amongst the various features that determine the successful landing?
⍰ What are the different conditions for successful landing of the rocket?

•

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Data was from SpaceX API and web scrapping from Wikipedia.
- Perform data wrangling
- Encoding was applied to categorial features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
- • How to build, tune, evaluate classification models

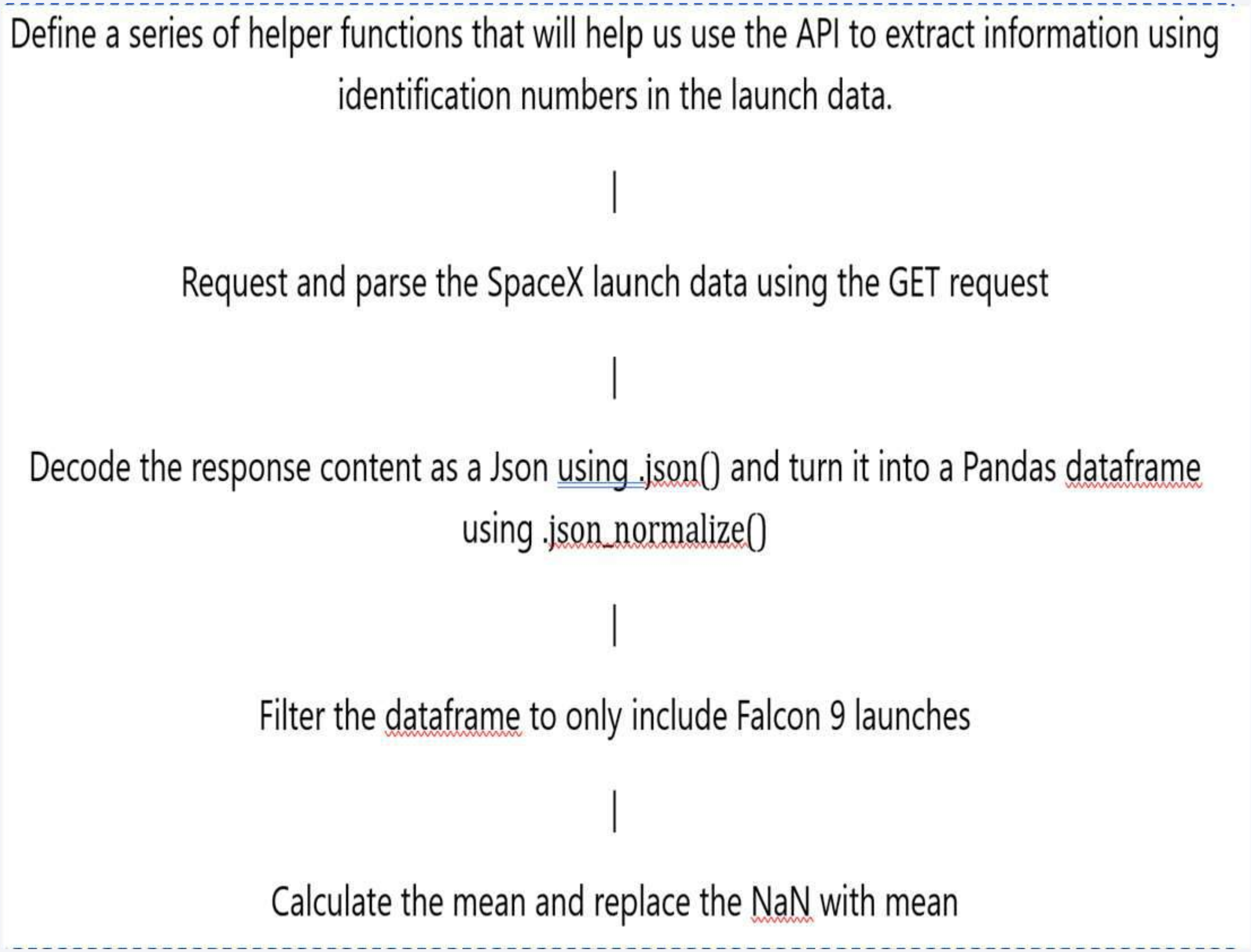
Data Collection

- The data was collected from various methods.
  - Data collection was done using get request to SpaceX API.
  - We decoded the response as Json using json() function and turn it to pandas dataframe using json_normalize().
  - We then checked the missing rows and filled the missing values where necessary.
  - In addition we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective is to extract the data, parse the table and convert it to a pandas dataframe for analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- Link : https://github.com/Pi3rr3git/DSCapstone/blob/main/jupyter-labs-spacex-data-collection-peMartin%20api.ipynb



Define a series of helper functions that will help us use the API to extract information using identification numbers in the launch data.

Request and parse the SpaceX launch data using the GET request

Decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()

Filter the dataframe to only include Falcon 9 launches

Calculate the mean and replace the NaN with mean

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

Place your flowchart of web scraping here

Data Wrangling

- We performed exploratory data analysis and determined the training labels.

- We calculated the number of launches of each site and the number of occurrence of each orbits.

- We also calculated the succuss rate of landing by finding the mean.

- We can also export the output to csv.

- Link - https://github.com/Pi3rr3git/DSCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling-peMartin.ipynb

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit, flight number and orbit type, payload and orbit type and finally the success rate yearly trend.

- Link - https://github.com/Pi3rr3git/DSCapstone/blob/main/jupyter-labs-peMartin.ipynb

EDA with SQL

- We load the dataset in our jupyter notebook.
- We applied EDA with SQL to get insight of our data. We applied queries to find out:

The names unique launch sites in the space mission.
Display top 5 records where launch site name beginning with particular string.
Total and Average payload mass carried by boosters.
Date of 1st successful landing in ground.
Total number of successful and failure missions and many more.

- Link - https://github.com/Pi3rr3git/DSCapstone/blob/main/jupyter-labs-peMartin-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We marked all the launch site on world map.
- We added circles to find the success and failure launches from each sites. For that we use color marker. Green color shows the successful landing outcome whereas Red color shows the failure in the mission.
- We also show the distance from the launch sites to its close landmarks like city, highway, railway station.
- We observe that all the sites are near the coasts and away from populated areas.
- Link - https://github.com/Pi3rr3git/DSCapstone/blob/main/lab_jupyter_launch_site_location%20peMartin.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash.
- We plotted pie charts showing the total launches by certain sites.
- We plotted the scatter graph showing the relationship with Outcome and Payload Mass for different booster version.
- Link - https://github.com/Pi3rr3git/DSCapstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- - We loaded the data using pandas and numpy, and divided the data into training and test sets.
- - We built different machine learning models.
- - We used accuracy as a metric for our model and improved the model using feature engineering.
- We improved the model using feature engineering.
- - We found the best performing classification model. We find that the decision tree model has the highest learning accuracy.
- - Link to https://github.com/Pi3rr3git/DSCapstone/blob/main/module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite-peMartin.ipynb

Results

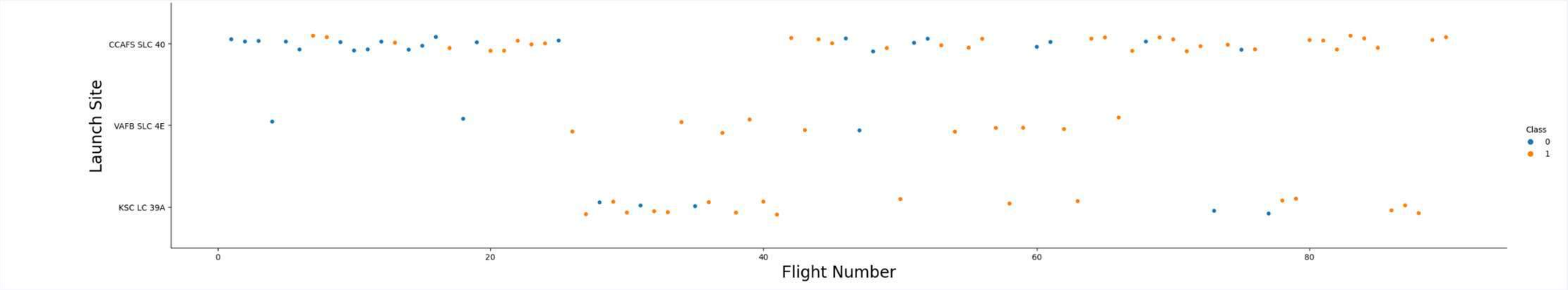
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

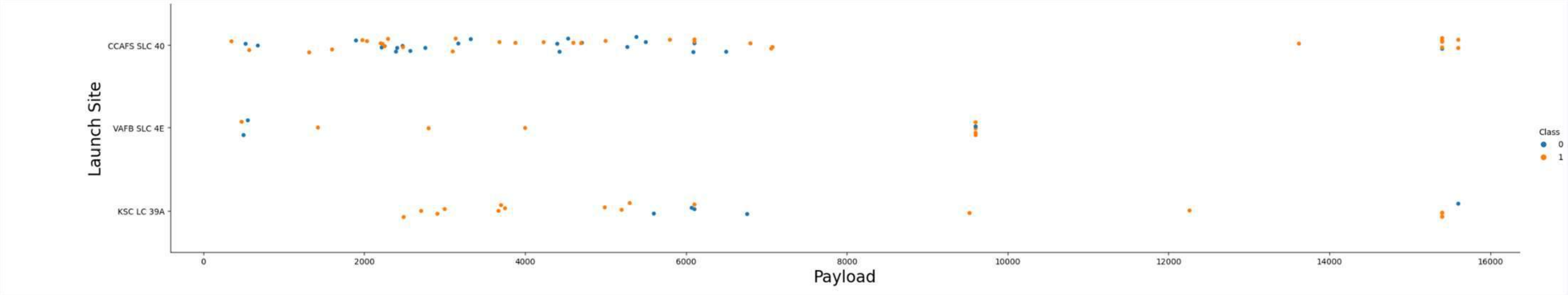
Flight Number vs. Launch Site

From the plot, we find that the larger the amount of flights, at a particular launch site, the greater is the success rate. Here CCAFS-SLC 40 has the maximum success rate.



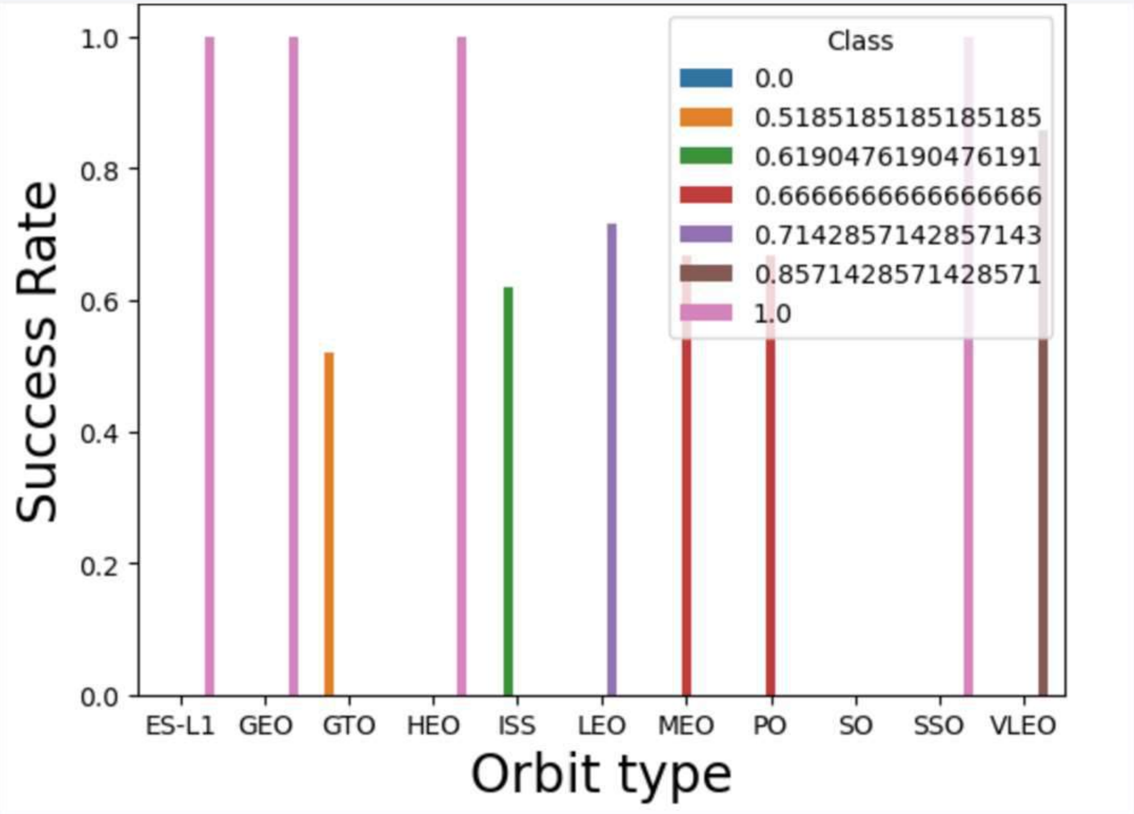
Payload vs. Launch Site

- From the plot, we find that VAFB SLC 4E launch site, there re no rockets of heavy payload mass greater than 10000.
- Also CCAFS SLC 40 has high success for high payload mass around 15000.



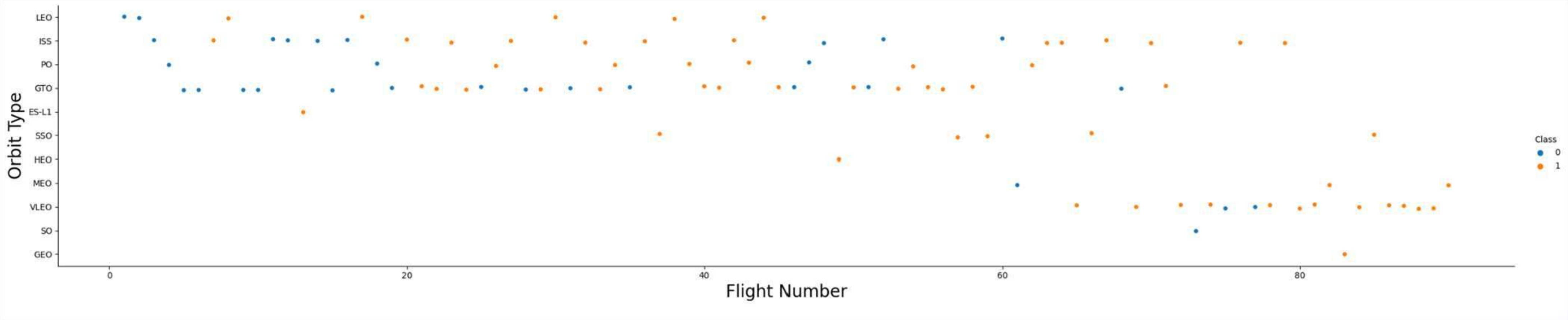
Success Rate vs. Orbit Type

- From the bar plot, we can clearly see that ES-L1, GEO, HEO and SSO have 100% success rate followed by VLEO.
- SO has 0 success rate.



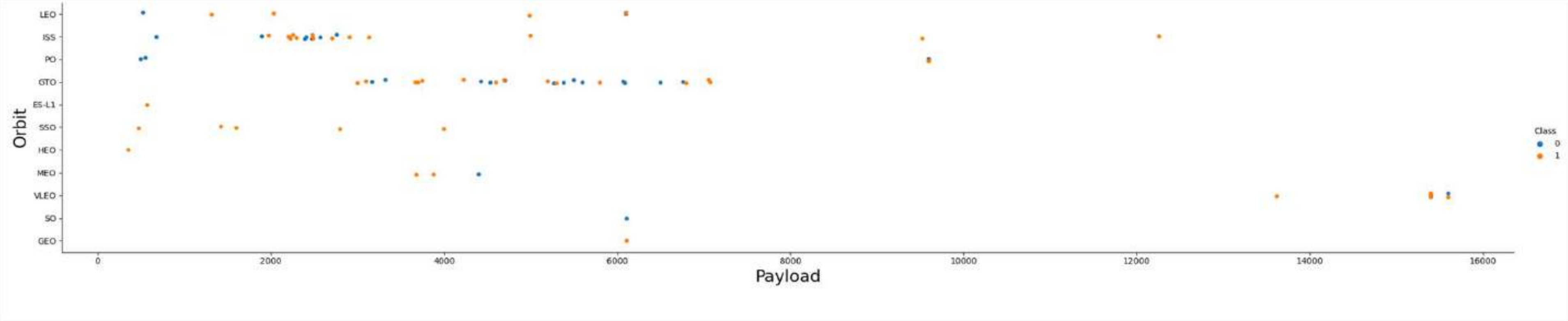
Flight Number vs. Orbit Type

- From the plot, we see that there are very few flights flown from LEO orbit. However all the flights are successful. On the other hand, there seems to be no relationship between flight number when in GTO orbit.



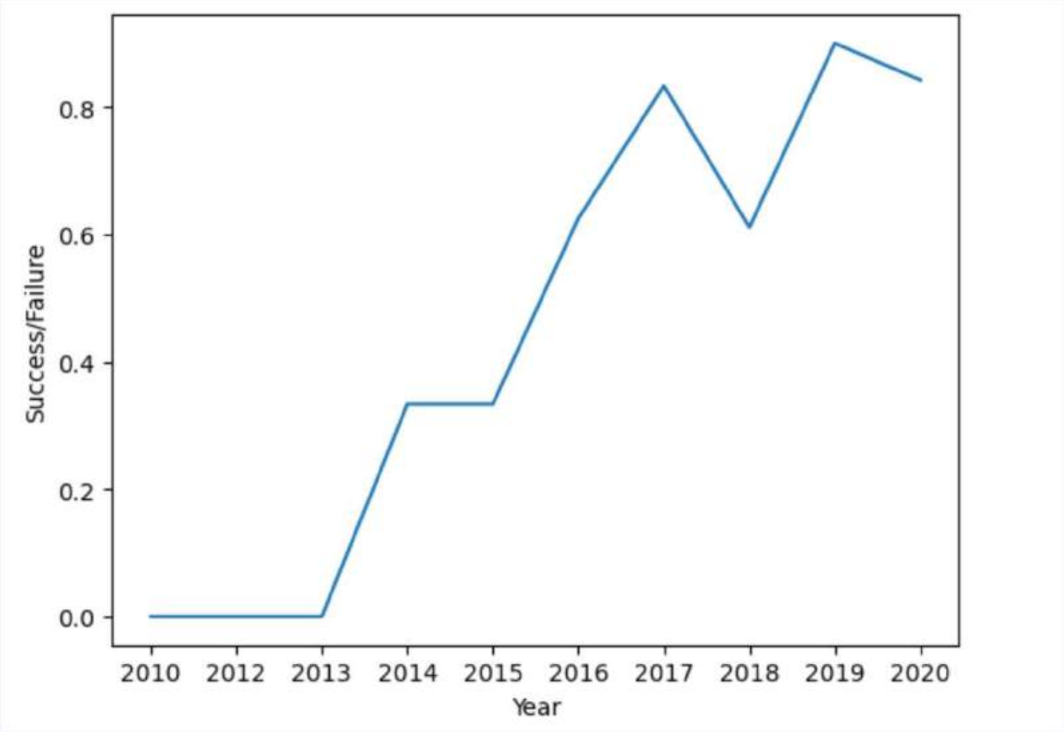
Payload vs. Orbit Type

- From the plot, we see with heavy payloads the successful landing rates are more for Polar, LEO and ISS.
- However for GTO, we cannot distinguish this as both positive and negative landing are there here.



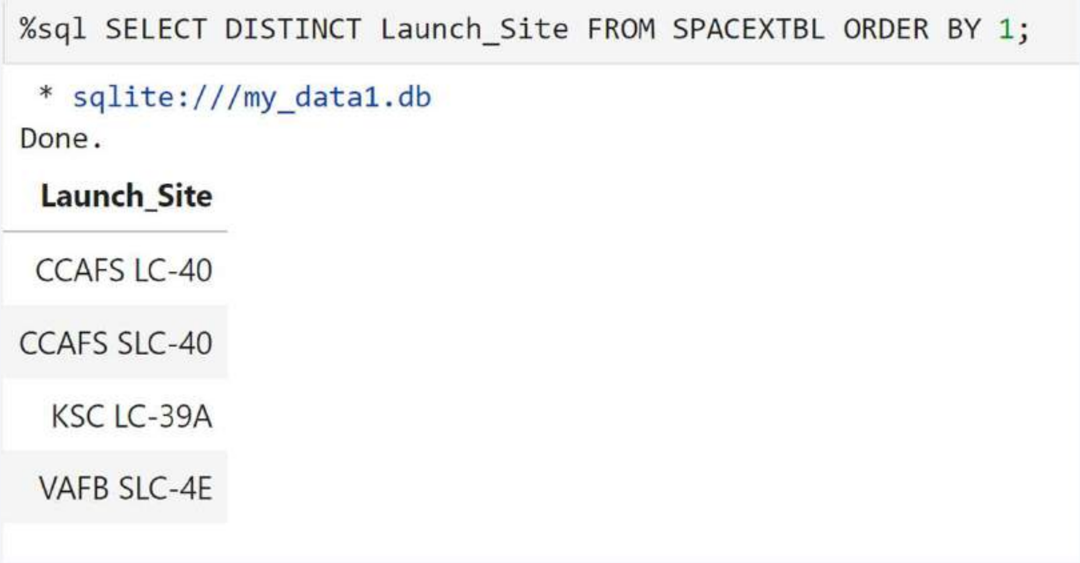
Launch Success Yearly Trend

- From the plot, we can see the success rate is increased in 2013 and it is constant till 2015, after that it boosted up till 2017.
- In mid of 2018 we can see sudden decrease which again increased 2019.



All Launch Site Names

We use the keyword ‘DISTINCT’ to find out unique launch sites for the mission.



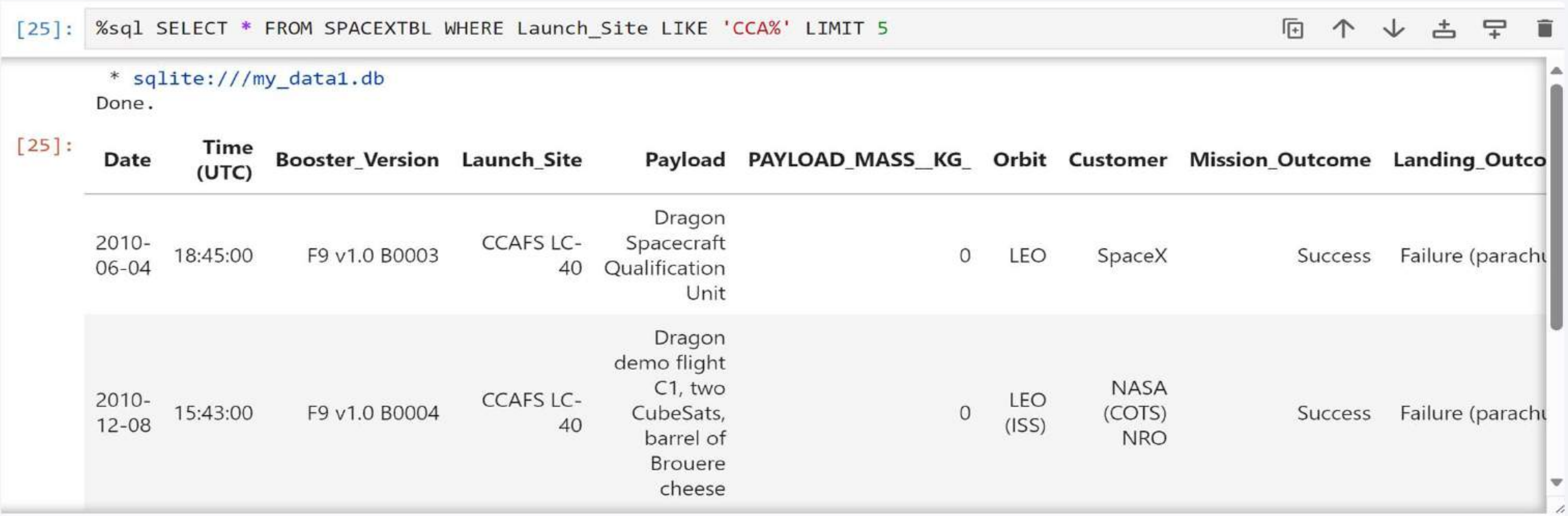
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL ORDER BY 1;
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- We use the ‘LIKE’ keyword to find the launch site names starting with ‘CCA’ and ‘LIMIT’ to find 1st 5 sites.



```
[25]: %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

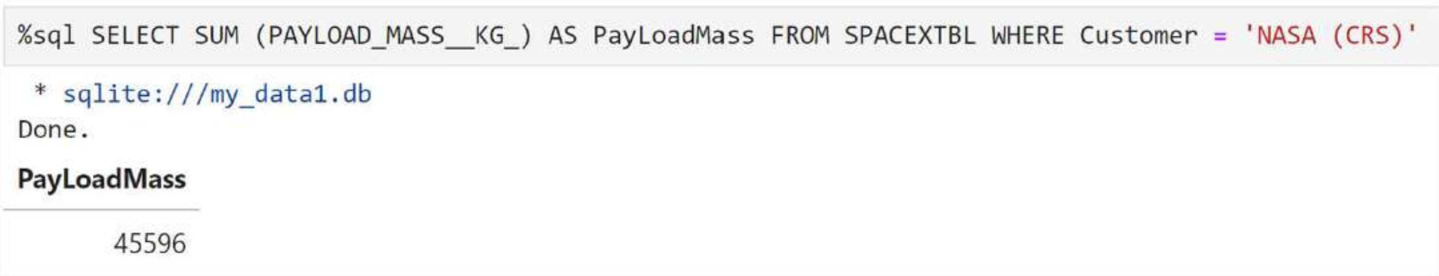
```
 * sqlite:///my_data1.db
Done.
```

[25]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outco
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachu
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 (parachu

Total Payload Mass

- We use the alias ‘AS’ keyword to change the column name of Payload mass.
- We also use ‘WHERE’ keyword to filter out the Customers.
- We also use ‘SUM’ for adding all the Payload Mass values.



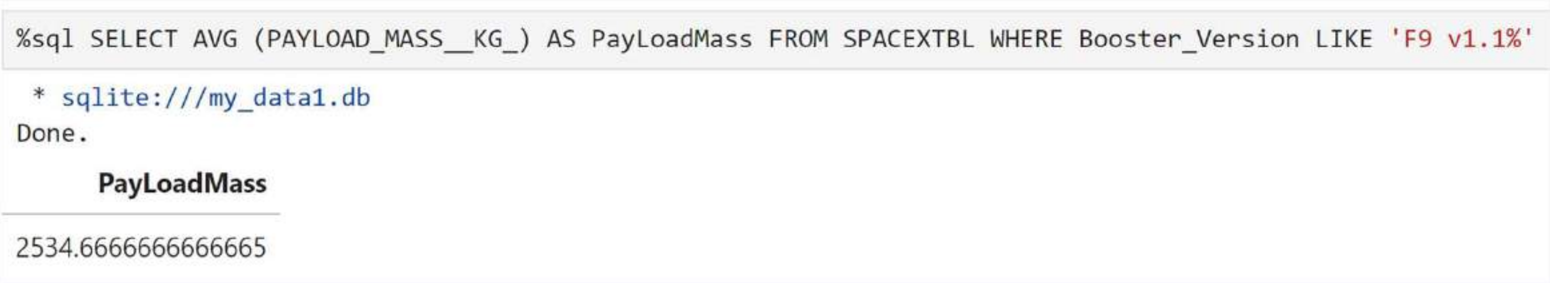
```
%sql SELECT SUM (PAYLOAD_MASS__KG_) AS PayLoadMass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

PayLoadMass
45596

Average Payload Mass by F9 v1.1

- We use ‘LIKE’ here again as there are many boosters starting with F9 v1.1 and ‘AVG’ to calculate the average of the Payload Mass.



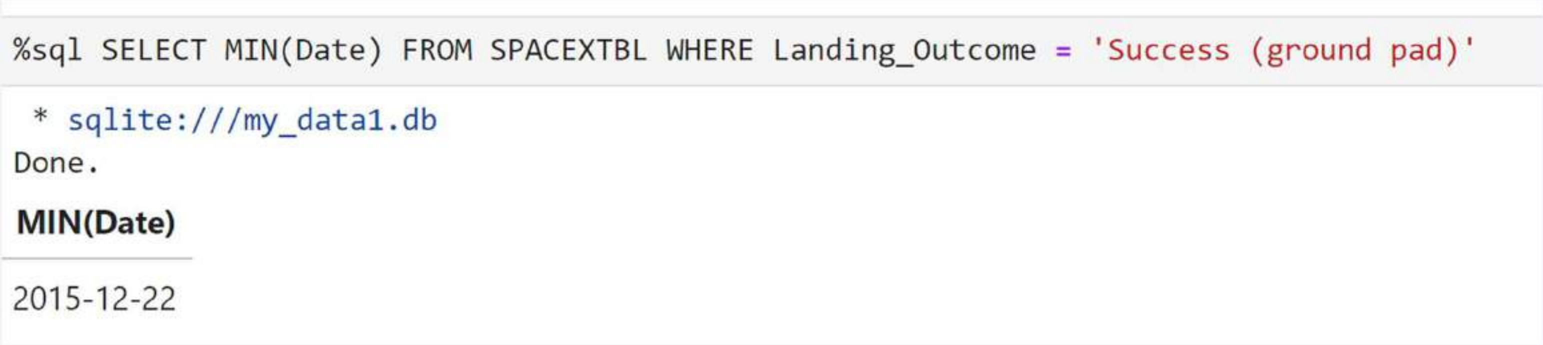
```
%sql SELECT AVG (PAYLOAD_MASS__KG_) AS PayLoadMass FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

```
 * sqlite:///my_data1.db
Done.
```

PayLoadMass
2534.6666666666665

First Successful Ground Landing Date

- We use the ‘MIN’ keyword to find the first successful mission of 2015

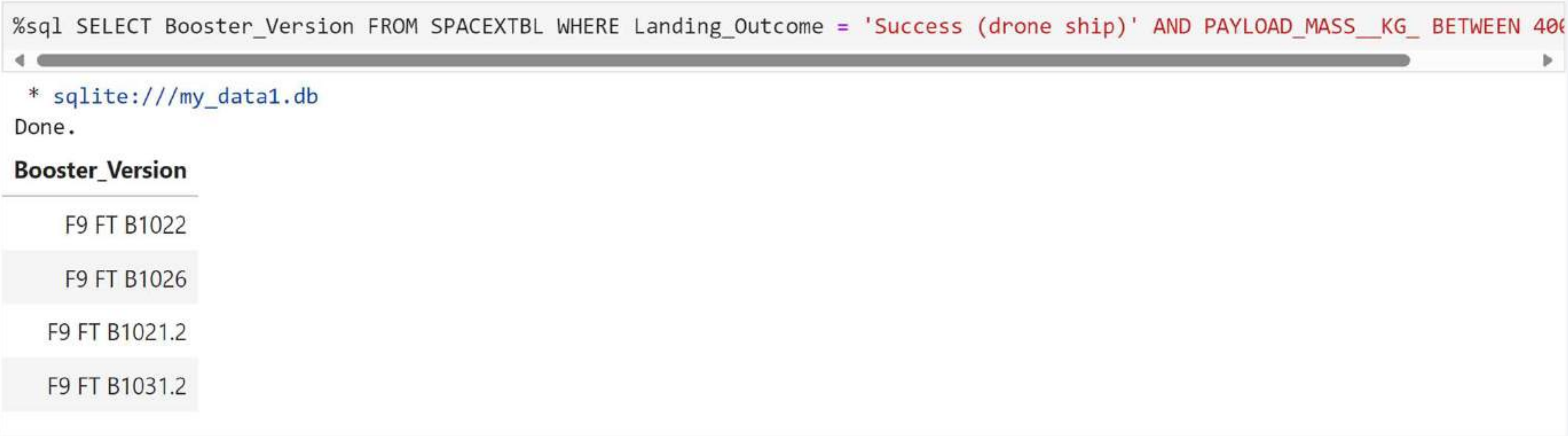


```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
MIN(Date)
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- We use the ‘BETWEEN’ keyword to filter out the Payload mass between 4000 and 6000.



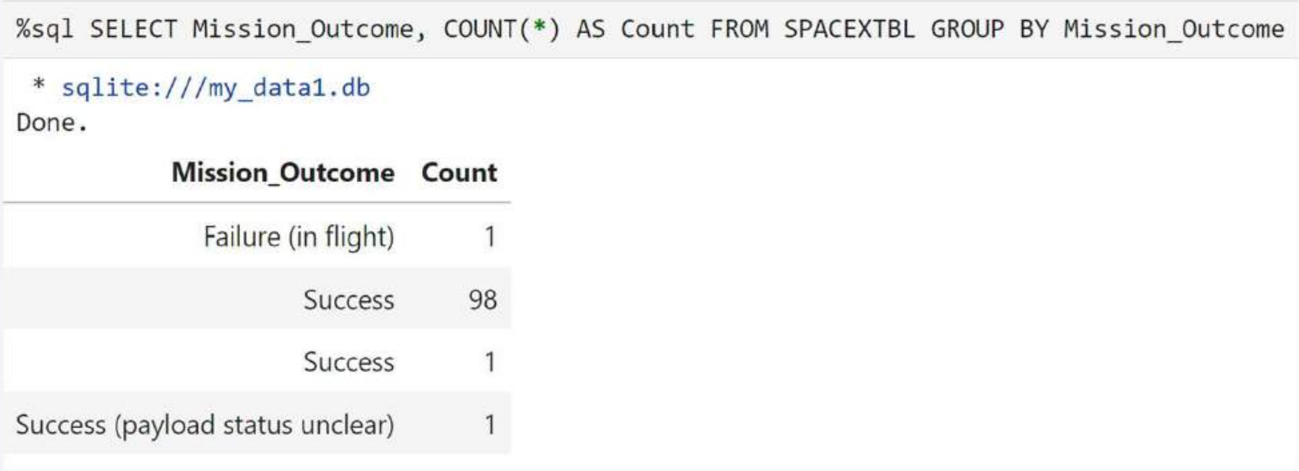
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 40
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We use the ‘COUNT’ keyword to count the total success and failure mission.
- We use ‘GROUP BY’ keyword to group all the mission outcomes together.



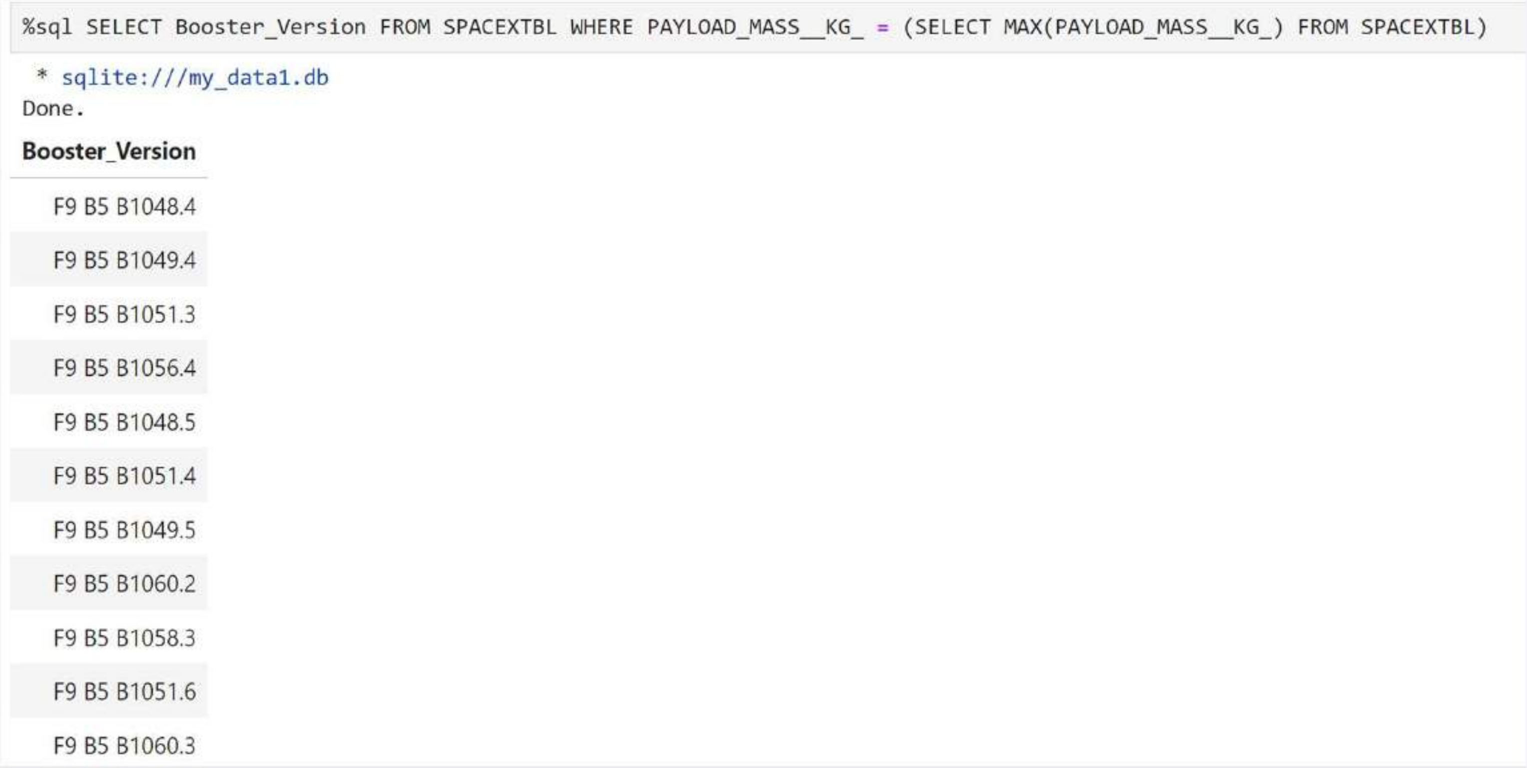
```
%sql SELECT Mission_Outcome, COUNT(*) AS Count FROM SPACEXTBL GROUP BY Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We use sub query to find maximum Payload.



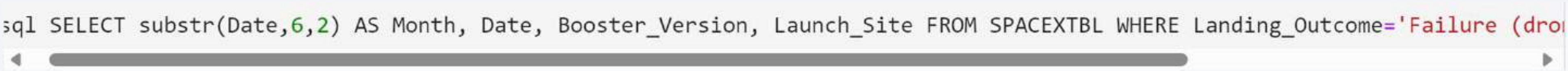
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

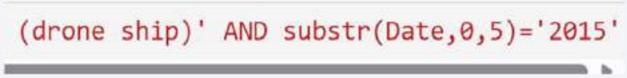
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3

2015 Launch Records

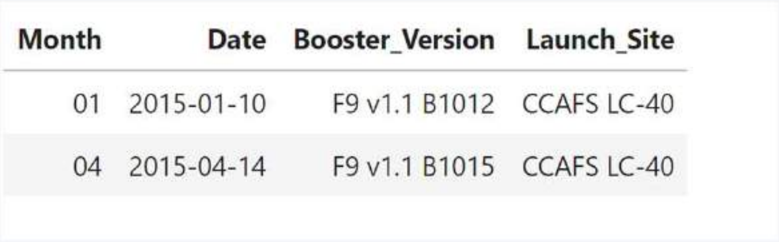
- Here we make use of substring to extract required data.



```
sql SELECT substr(Date,6,2) AS Month, Date, Booster_Version, Launch_Site FROM SPACEXTBL WHERE Landing_Outcome='Failure (dro
```



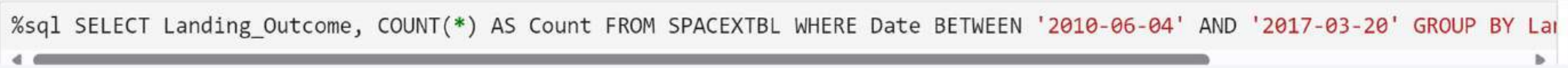
```
(drone ship)' AND substr(Date,0,5)='2015'
```



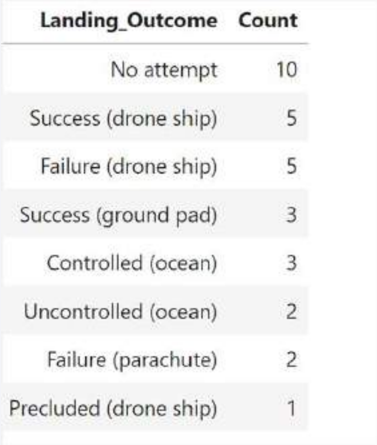
Month	Date	Booster_Version	Launch_Site
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40

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

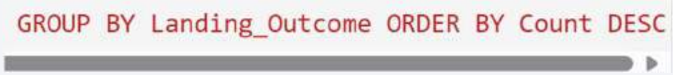
- We use the ‘GROUP BY’ and ‘ORDER BY’ clause to group all the even records and sort them. Furthur we use ‘DESC’ to sort in descending order.



```
%sql SELECT Landing_Outcome, COUNT(*) AS Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY La
```



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

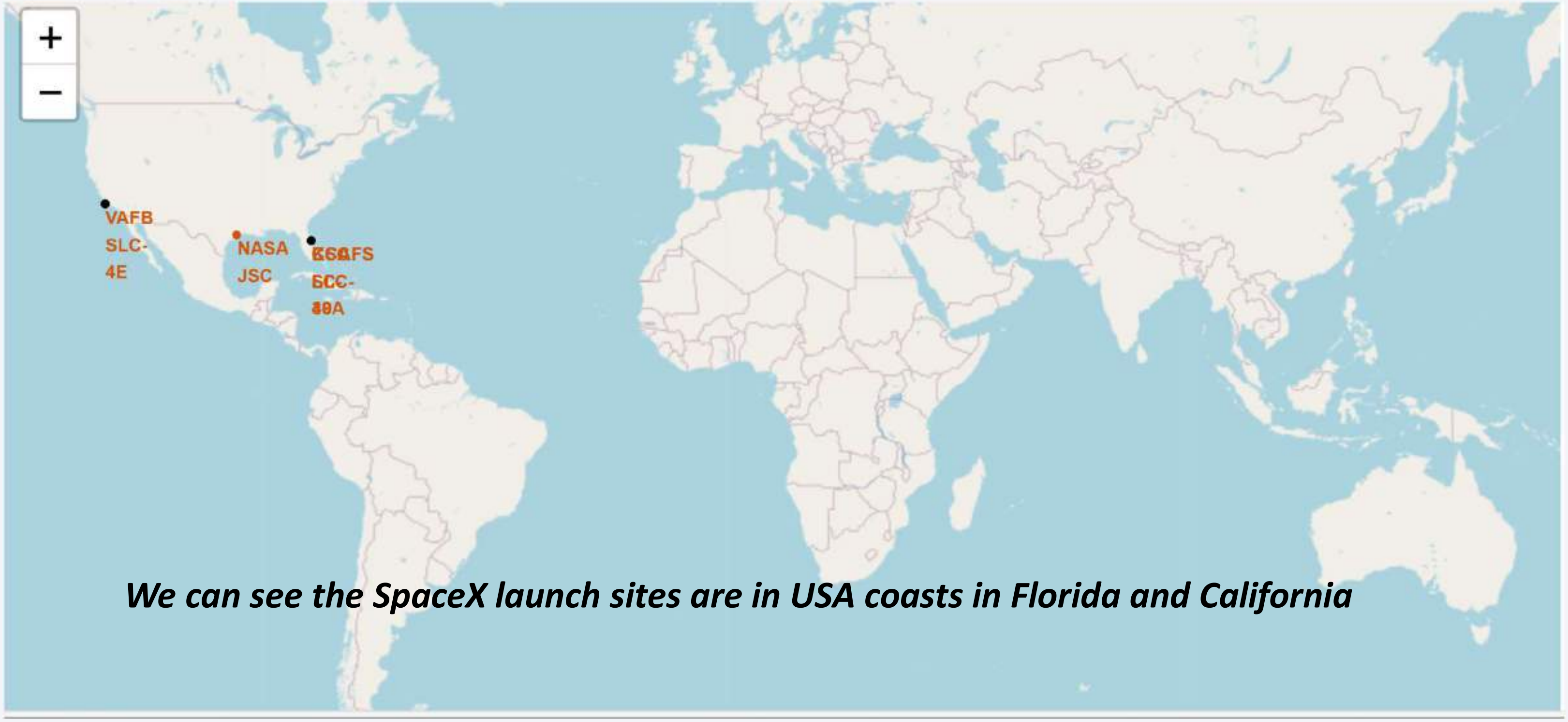


```
GROUP BY Landing_Outcome ORDER BY Count DESC
```

Section 3

Launch Sites Proximities Analysis

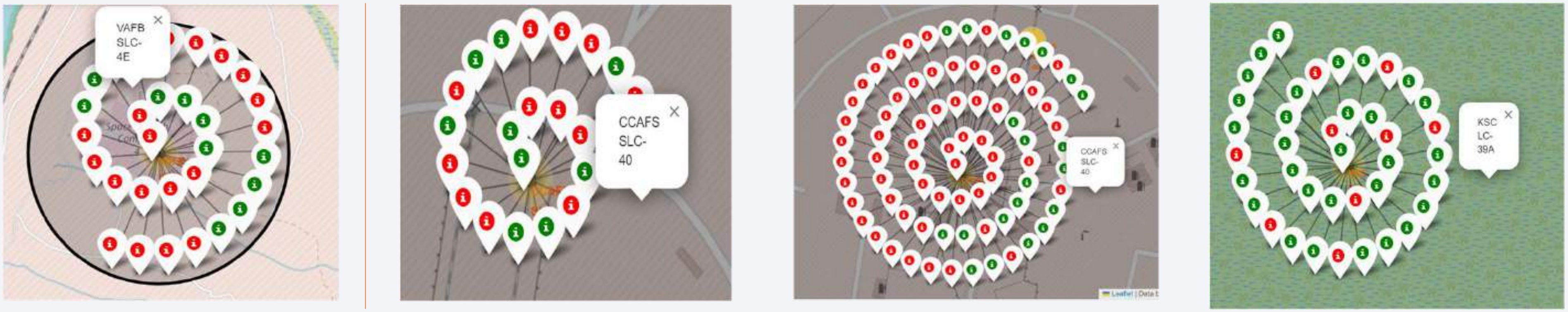
All launch sites on global map



We can see the SpaceX launch sites are in USA coasts in Florida and California

Markers showing launch sites with color labels

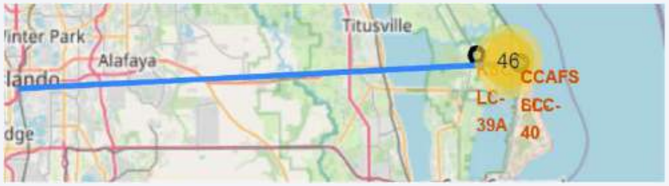
- VABSLC-4E launch site is the only site on the west coast of USA, whereas others are on the eastern coast.

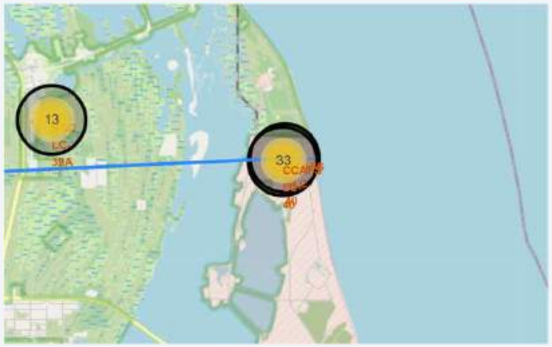


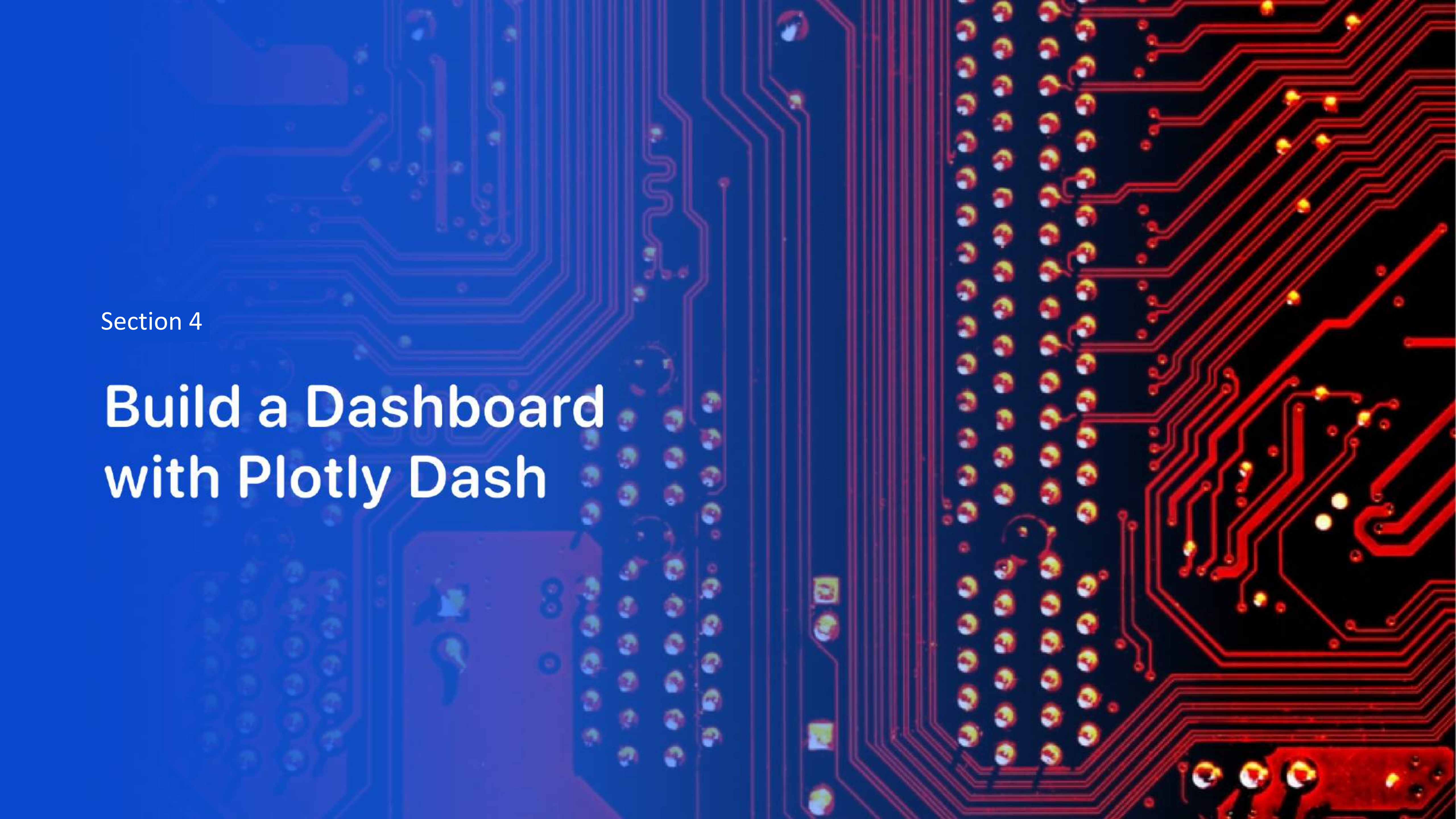
Green marker shows successful launches and Red marker shows Failures

Launch Site distance to landmarks

- So if we see, the launch sites are near the coasts.
- The nearest city which is Orlando, is far away from the 2 launch sites on the eastern coast.
- Also highways and railway lines are far away from the sites.



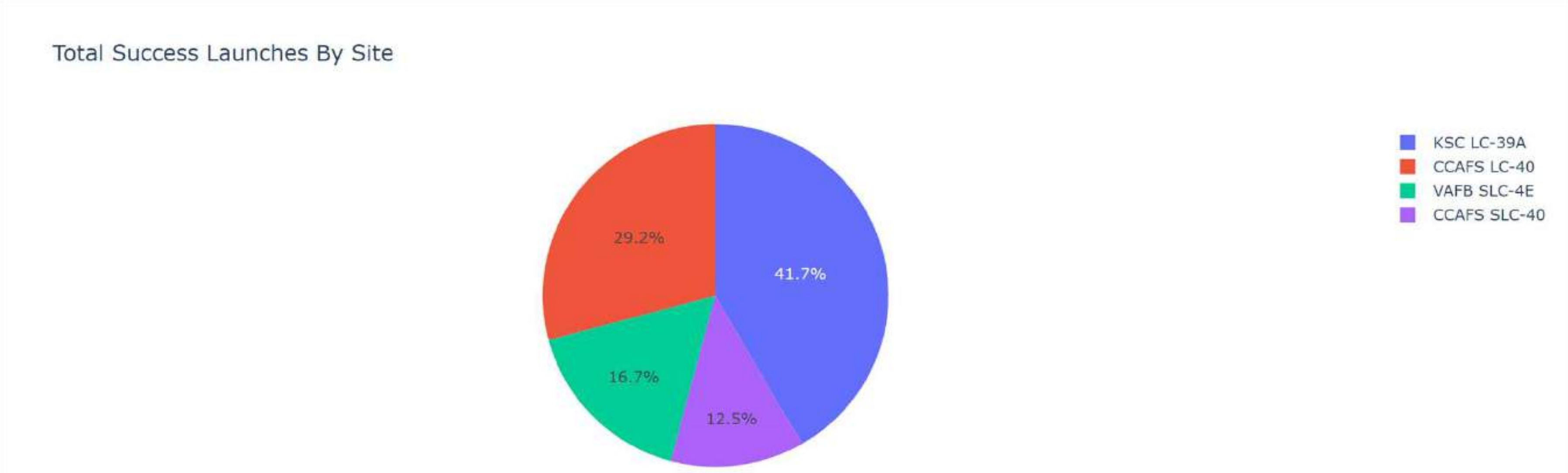




Section 4

Build a Dashboard with Plotly Dash

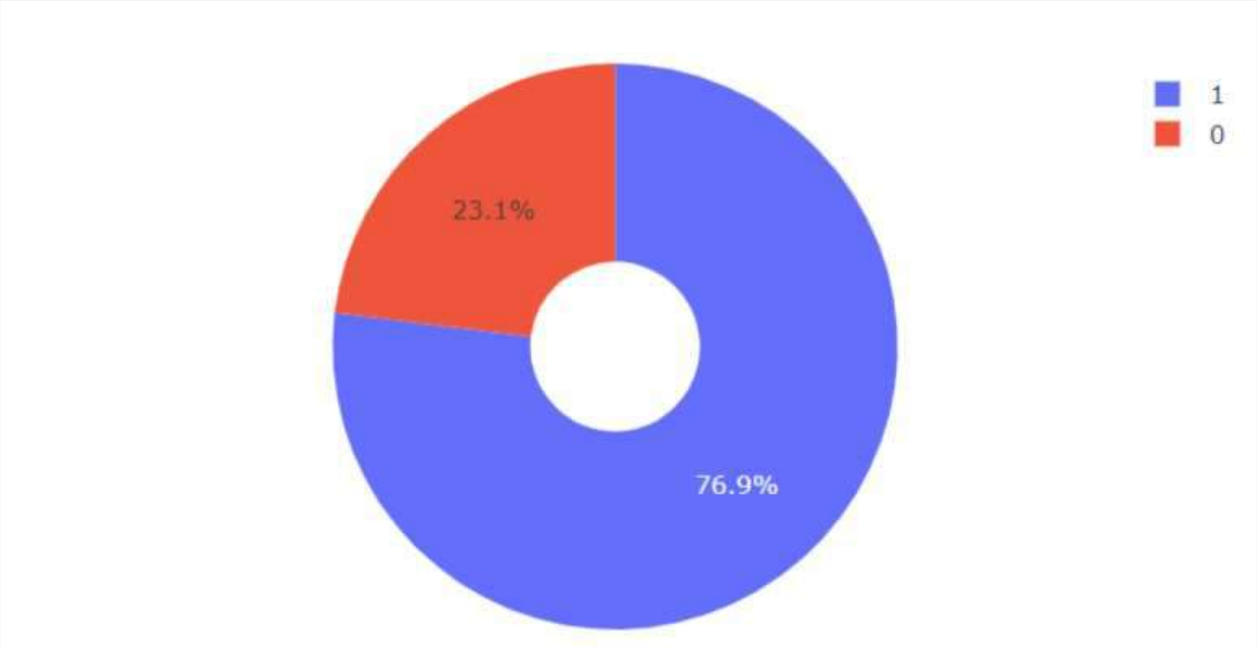
Pie chart showing the success percentage achieved by each launch site



We can see that KSC LC-39A had the most successful launches from all the sites

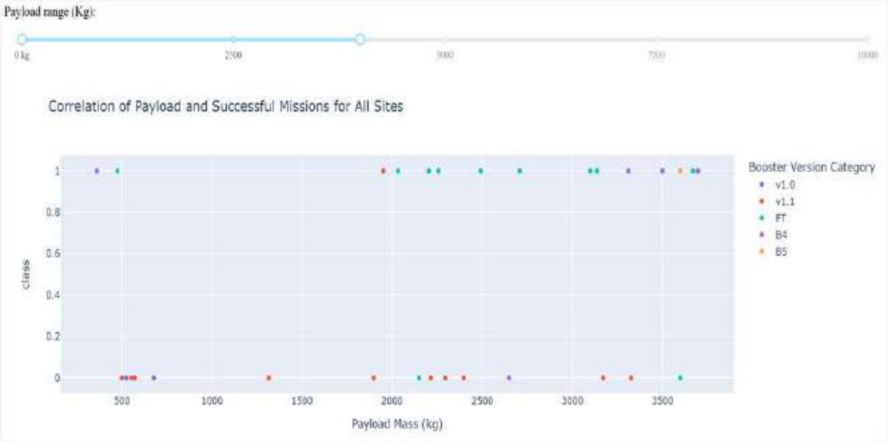
From the plot we can see that KSC LC -39A has the most successful launches compared to all other sites

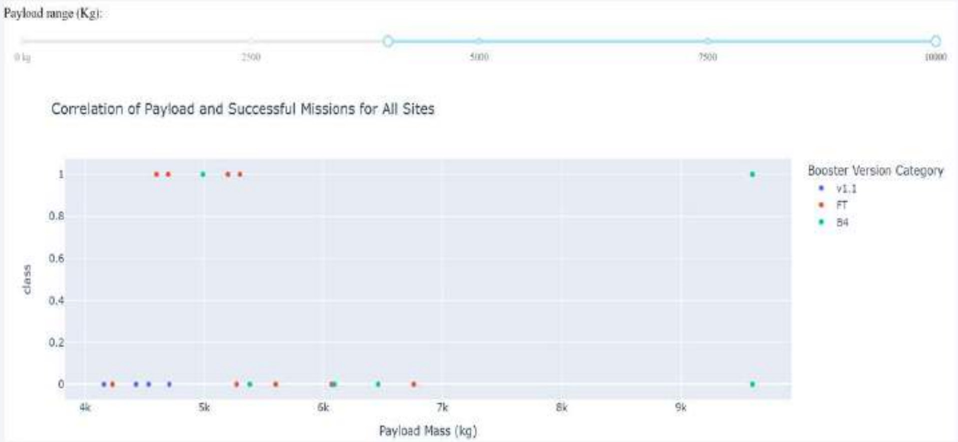
Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





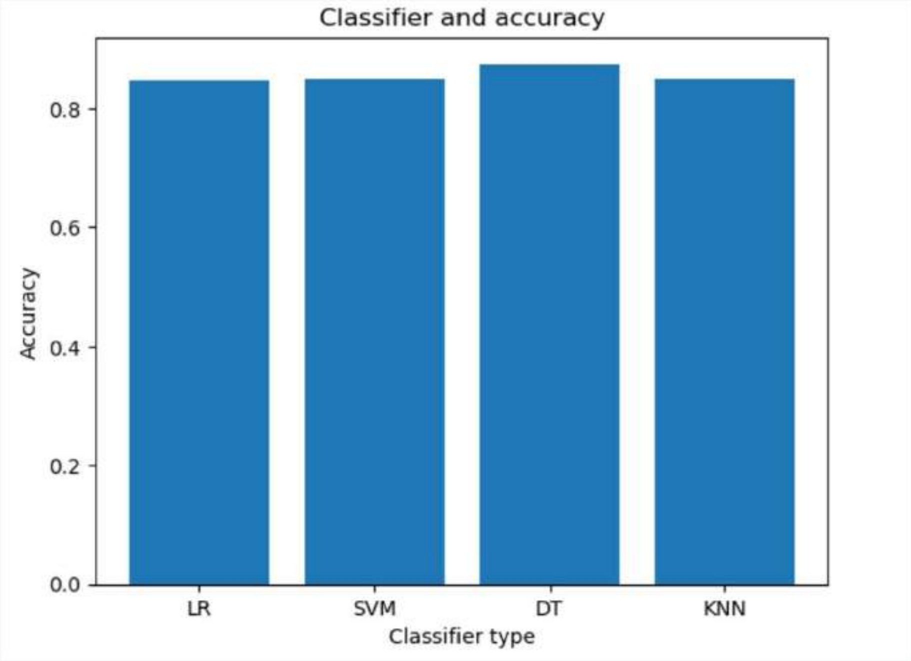
We see that the success rates of low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

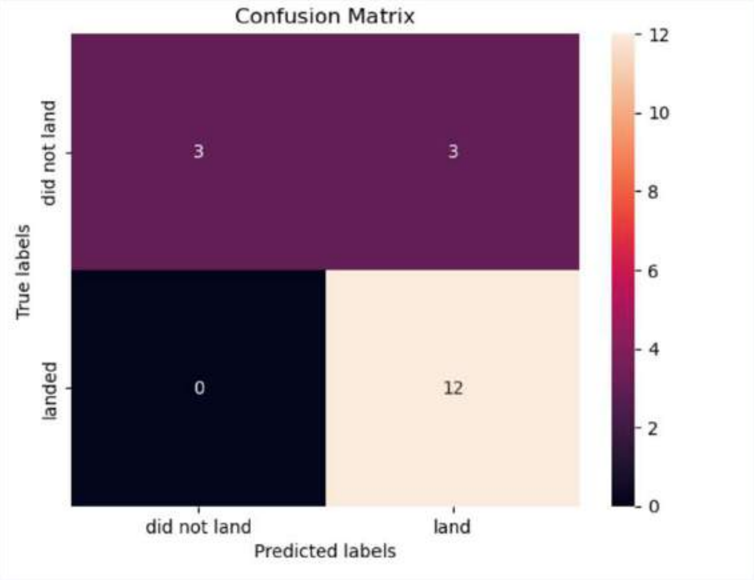
Classification Accuracy

- From the barplot, we can classify the accuracy of training sets of different classifiers.
- We can see that the accuracy of accuracy of all the models is almost the same around 83%, however the accuracy of Decision Tree model is higher that others, so in this case Decision Tree is most accurate classifier.



Confusion Matrix

- The confusion matrix for Decision Tree shows that the classifier can distinguish between different classes.
- The false positive is a problem i.e the chance of successful landing marked as successful by the classifier.



Conclusions

We conclude that:

- Decision Tree model is the best algorithm for this dataset.
- The larger the flight amount at the launch site, the greater is the success rate at a launch site.
- Launch success for low payload mass is better for all the orbits.
- Launch success rate started to increase as the years went past.
- Orbits ES L1, GEO, HEO, SO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- Launch sites are placed near coastal areas and awat from populated areas.

Thank you!